UNICESUMAR

Luis Hamilton Balem

TRABALHO DESAFIO PROFISSIONAL III

Curitiba

2023

Luis Hamilton Balem

Trabalho Prático

Montagem de Um Ambiente Virtual Web Vulnerável

Relatório

UNICESUMAR

Professora Ana Paula

Curitiba/PR, 23 de Junho de 2023

Sumário

1. Introdução

1.1 Contexto e Justificativa do Trabalho Prático

O trabalho consiste na montagem de um ambiente virtual web vulnerável, que simula uma aplicação real com diversas vulnerabilidades conhecidas. Montar um ambiente virtual web vulnerável permite um aprendizado prático, sensibilização sobre segurança, preparação para defesa, melhoria das habilidades de desenvolvimento seguro e testes e validação de ferramentas de segurança.

1.2 Objetivos

O objetivo desse trabalho é proporcionar um ambiente de aprendizado que permite aos participantes explorar e entender as vulnerabilidades comuns em aplicações web. Isso inclui identificar e compreender as principais vulnerabilidades em aplicações web, explorar as técnicas de exploração de vulnerabilidades, compreender os riscos associados às vulnerabilidades, aprender a implementar medidas de mitigação e prevenção, promover a conscientização sobre segurança cibernética e desenvolver habilidades de detecção e mitigação de vulnerabilidades. Para atender aos requisitos será criado, configurado uma máquina virtual que execute o sistema operacional Linux.

1.3 Metodologia

Para a prática desse trabalho será utilizado VirtualBox como máquina virtual, Kali Linux como sistema operacional escolhido para ser utilizado na máquina virtual. E o SQL Injection foi escolhido como ferramenta de estudo como vulnerabilidade dentro do ambiente virtual do WebGoat.

2. Ambiente Virtual

2.1 Instalação e Configuração do VirtualBox

- Faça o download do VirtualBox no site oficial;
- Execute o instalador e siga as instruções do assistente de instalação;
- Após a instalação, abra o VirtualBox;
- Clique em “Novo” para criar uma nova máquina virtual;
- Defina um nome, escolha o tipo e a versão do sistema operacional;
- Atribua a quantidade de memória RAM desejada para a máquina virtual;
- Selecione a opção de criar um disco rígido virtual e defina o tipo e tamanho do disco;
- Inicia a máquina virtual e selecione o arquivo de imagem do sistema operacional para a instalação;
- Siga as instruções do sistema operacional para concluir a instalação;
- Após a instalação, sua máquina virtual estará pronta para uso.

Durante o trabalho prático essa etapa foi realizada sem demais dificuldades, a orientação fornecida foi o suficiente, mesmo que nesse trabalho tenha sido realizado algumas configurações adicionais por escolha própria.

2.2 Instalação e Configuração do Linux na Máquina Virtual

- Obtenha a imagem ISSO do sistema operacional Linux que você deseja instalar;
- Você pode baixar a imagem ISO do site oficial;
- Abra o VirtualBox e clique em “Novo” para criar uma nova máquina virtual;
- Passe por alguns processos de configuração da sua máquina virtual;

- Na janela de inicialização, selecione a imagem ISO do Linux que você baixou como arquivo de inicialização;
- Inicie a máquina virtual e siga as instruções do assistente de instalação do Linux para instalar o sistema operacional na máquina virtual;
- Durante a instalação você precisa fornecer informações como idioma, layout do teclado, nome de usuário e senha;
- Após a conclusão, reinicie a máquina virtual e faça login com as credenciais do usuário que você criou;
- Agora, você tem o Linux instalado em sua máquina virtual.

Nessa etapa a primeira dificuldade encontrada foi para baixar a imagem do sistema operacional, já que o seu tamanho é razoavelmente grande a internet móvel não teve rapidez em baixar o programa. Após isso foi perdido algum tempo em pesquisa na internet em busca de como acessar uma sessão do sistema operacional, para descobrir qual login e senha padrão que já estavam sendo configurados automaticamente.

2.3 Instalação e Configuração do WebGoat

- No seu computador, baixe a versão mais recente do WebGoat da OWASP e salve o arquivo;
- Dentro da máquina virtual: instale o Java Runtime Environment (JRE) no Linux usando o seguinte comando no terminal: *sudo aptget install default-jre;*
- Instale o WebGoat no Linux usando o seguinte comando no terminal*: wget "url da versão mais recente do WebGoat";*
- Abra um navegador da web no Linux e acesse: *http://localhost:8080/WebGoat/* para acessar o WebGoat;
- Inicie o servidor Java através do seguinte comando: *java -jar webgoat-2023.4.jar*

Nessa etapa o problema encontrado foi atualizar o link fornecido na instrução por causa da diferença de versão e formatação do link, rapidamente resolvido, mas em outros trabalhos é muito provável que resultou em maiores dificuldades.

3. Visão Geral do WebGoat

3.1 Descrição e Funcionalidades do WebGoat

O WebGoat é uma aplicação web desenvolvida pela Open Web Application Security Project (OWASP) com o objetivo de ajudar os desenvolvedores de software a aprenderem sobre vulnerabilidades em aplicações web e como corrigi-las. Ele é uma ferramenta de treinamento para prática de testes de penetração (pentest) em aplicações web em um ambiente virtual.

O WebGoat possui diversos laboratórios que ensinam sobre vulnerabilidades em aplicações web, como SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), entre outras. Cada laboratório possui uma explicação sobre a vulnerabilidade e um ambiente para praticar a exploração da vulnerabilidade.

3.2 Como Acessar e Navegar no WebGoat

É possível notar que a URL do WebGoat se inicia com HTTP, ou seja, ele não é um site seguro, para evitar problemas é necessário criar uma máquina virtual com sistema operacional Linux, após realizar todos os processos e estar logado no sistema Linux, é necessário ter um servidor Java em execução. Quando a situação do servidor estiver nos conformes, basta acessar no navegador o IP do servidor seguido da porta do WebGoat, como: *http://localhost:8080/WebGoat* .

Na página inicial do WebGoat, é possível ver uma lista de desafios e exercícios relacionados a vulnerabilidades de segurança web. Ao selecionar um desafio, deve-se seguir as instruções fornecidas pelo WebGoat para explorar uma vulnerabilidade específica. O WebGoat fornecerá orientações e dicas para auxiliar durante o processo de aprendizado.

4. Práticas Comuns de Segurança em Aplicações Web

4.1 Conceitos Básicos de Segurança em Aplicações Web

A segurança em aplicações web é importante por diversos motivos, mas um dos principais é que a maioria das empresas coleta, armazena e utiliza informações confidenciais de seus clientes, tais como dados pessoais, financeiros e de transações. Essas informações são um alvo para atacantes, que buscam explorar vulnerabilidades para obter acesso a dados sensíveis e causar danos significativos à empresa e seus clientes.

4.2 Identificação de Vulnerabilidades Comuns em Aplicações Web

Injeção de código: Isso ocorre quando um aplicativo web não valida corretamente a entrada do usuário e permite a inserção de código malicioso, como SQL Injection ou XSS (Cross-Site Scripting), esse tipo de vulnerabilidade (XSS), os invasores inserem scripts maliciosos nos dados fornecidos pelo usuário, que são então executados no navegador de outros usuários, comprometendo sua segurança.

Cross-Site Request Forgery (CSFR), acontece quando um invasor induz um usuário autenticado a executar ações indesejadas em um aplicativo web sem seu conhecimento ou consentimento, explorando a confiança do aplicativo na identidade do usuário.

Configuração incorreta do servidor web, banco de dados ou outros componentes podem abrir brechas de segurança. Por exemplo, configurações padrão não alteradas, permissões excessivas ou não autenticadas. Exposição de informações sensíveis, quando informações confidenciais como senhas, são armazenadas ou transmitidas de forma insegura, e acabam sendo acessadas por invasores.

Autenticação e gerenciamento de sessões inadequados: Erros ao implementar autenticação de usuários e gerenciamento de sessões podem permitir ataques de força bruta, roubo de sessões ou escalonamento de privilégios.

Deserialização insegura, pode permitir que invasores modifiquem o conteúdo serializado para executar código malicioso ou explorar vulnerabilidades no processo de deserialização. Acesso não autorizado: Falhas de controle de acesso podem permitir que usuários não autorizados acessem recursos ou execute ações restritas.

4.3 Boas Práticas para Mitigação de Vulnerabilidades em Aplicações Web

Validação e filtro de entrada de dados para garantir integridade. Uso de parâmetros adequados ao construir consultas de banco de dados, utilize parâmetros preparados ou consultas parametrizadas em vez de concatenação de strings (variáveis de texto). Sanitização e escape de saída ao exibir dados fornecidos pelos usuários em páginas web para garantir que eles sejam exibidos corretamente e não sejam interpretados como código malicioso pelos navegadores.

Implementação correta de autenticação, hashes, armazenamento seguro de credenciais, autenticação de dois fatores, para fortalecer a segurança. Gerenciamento adequado de sessões, ou seja, implementação de token temporários exclusivos para cada usuário. Atribuição de permissões mínimas aos usuários e componentes do sistema, diminuindo brechas.

Atualização e patching, mantendo todas as bibliotecas, frameworks e componentes atualizados e com correções de segurança. Configuração segura do servidor, seguindo melhores práticas. Testes de segurança como método de prevenção e conscientização sobre novas práticas de segurança, promovendo uma cultura de segurança desde o início do ciclo de desenvolvimento.

4.4 SQL Injection

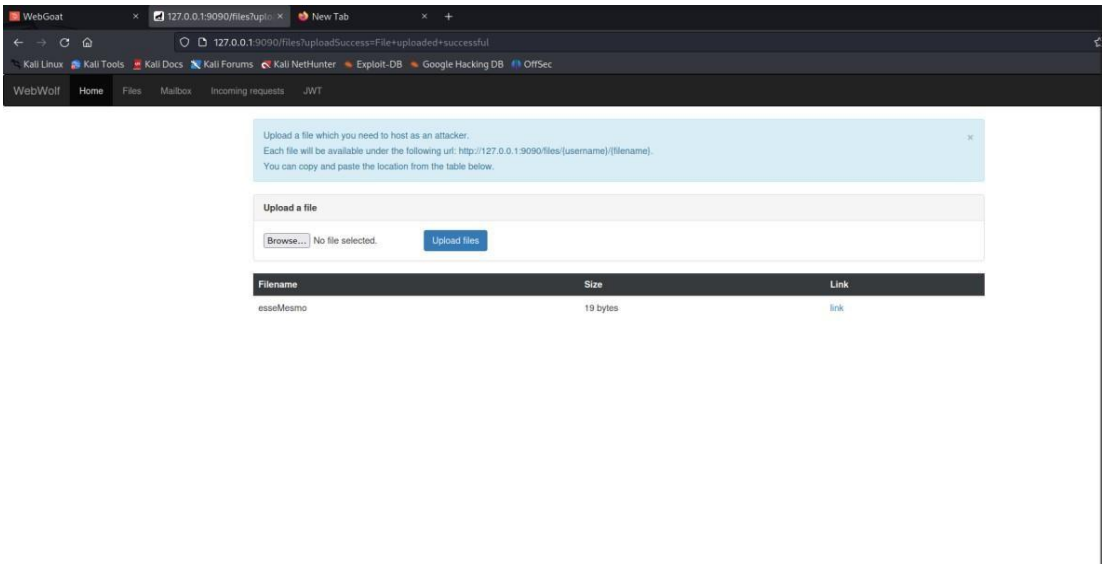
Uma técnica de ataque que explora falhas de segurança em sistemas que utilizam linguagem SQL para acesso a banco de dados. O objetivo é inserir códigos maliciosos na consulta SQL para executar ações não autorizadas, como modificar ou excluir dados, ou seja, os atacantes podem explorar o SQL Injection

para obter informações confidenciais ou executar comandos maliciosos no banco de dados.

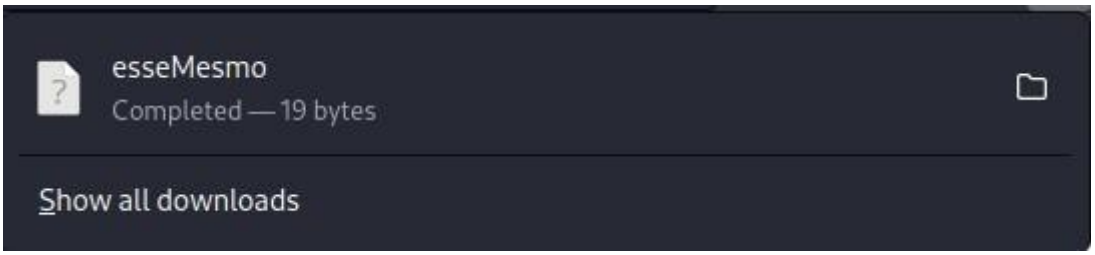
5. Conclusão

5.1 Síntese dos Resultados e Conclusões do Trabalho Prático

Foi feito upload de um arquivo “esseMesmo.txt” para o WebWolf

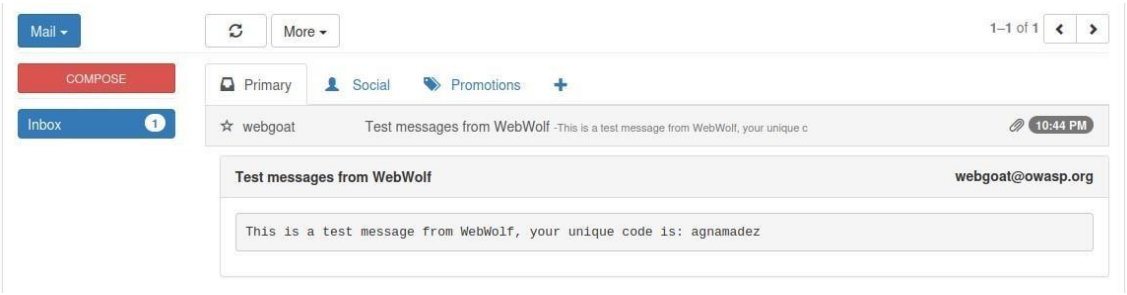


É possível baixar o arquivo através do link de upload gerado pelo WebWolf *http://127.0.0.1:9090/files/{username}/{filename}*.

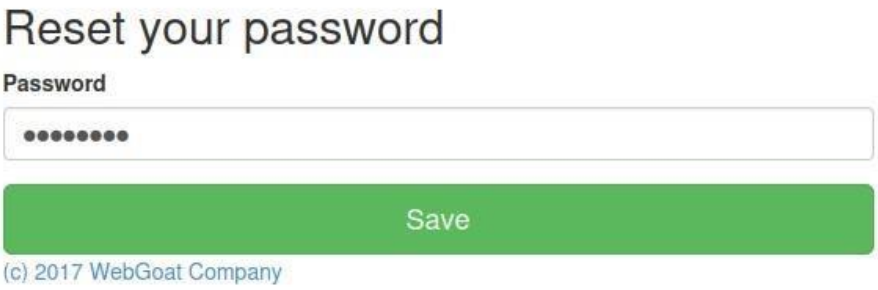


Através do WebGoat foi enviado um e-mail para o MailBox, um client de e-mails do WebWolf

Aqui mostra que o e-mail foi recebido



Aqui simulamos um usuário sendo enganado a mudar sua senha, acabando por mandá-la para o WebWolf nas requisições



Aqui podemos ver que foi recebida a requisição com a senha inclusa

2023-06-07T02:53:14.888837617Z | /landing


```
{
  "timestamp" : "2023-06-07T02:53:14.888837617Z",
  "principal" : null,
  "session" : null,
  "request" : {
    "method" : "GET",
    "uri" : "http://127.0.0.1:9090/landing?uniqueCode=agnamadez&password=KARAMELO",
    "headers" : {
      "Cookie" : [ "WEBWOLFSESSION=usitU0XLFKBfaJ2nDHU6iEjj4pJehxrFc-Pdao7r" ],
      "Accept" : [ "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8" ],
      "User-Agent" : [ "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0" ],
      "Connection" : [ "keep-alive" ],
      "Referer" : [ "http://localhost:8080/" ],
      "Sec-Fetch-Dest" : [ "document" ],
      "Sec-Fetch-Site" : [ "cross-site" ],
      "Host" : [ "127.0.0.1:9090" ],
      "Accept-Encoding" : [ "gzip, deflate, br" ],
      "Sec-Fetch-Mode" : [ "navigate" ],
      "Upgrade-Insecure-Requests" : [ "1" ],
      "Sec-Fetch-User" : [ "?1" ],
      "Accept-Language" : [ "en-US,en;q=0.5" ]
    },
    "remoteAddress" : null
  },
  "response" : {
    "status" : 200,
    "headers" : { }
  },
  "timeTaken" : 24
}
```


Após localizar o código único e mandá-lo no WebGoat aparece a seguinte mensagem:

Em conclusão, foram exercícios muito proveitosos, onde mostram um pouco de como funciona o perigo do Phishing e como ele atua.

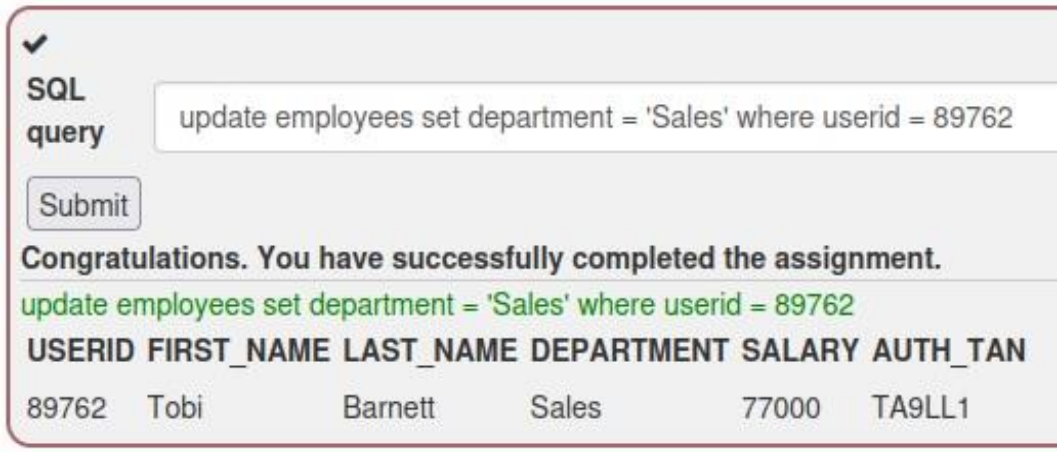
5.2 Limitações do Trabalho e Sugestões para Trabalhos Futuros

Como já abordados, os maiores limites/dificuldades foram na instalação e configuração do sistema operacional, além do acesso ao WebGoat. Para o desenvolvimento da próxima parte será desenvolvido as atividades sobre SQL Injection, presentes no WebGoat também, as quais compreendem a extração de informações de tabelas de dados, alteração e manipulação de permissões a credenciais de usuários. E por serem muito completas e explicativas, contribuem muito positivamente para o aprendizado.

6. Exercícios Parte 2

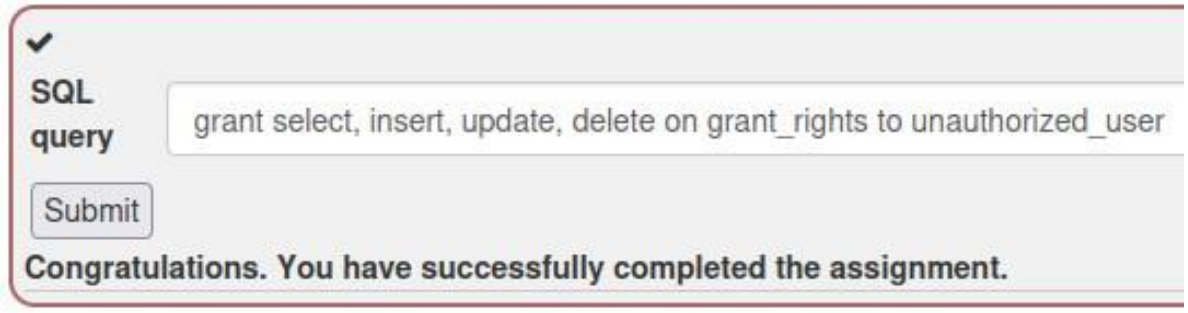
Nesse exercício o objetivo era coletar o departamento do empregado Bob Franco. Então eu fiz um comando de Select através do id do usuário. (*SELECT department FROM employees WHERE userid = 96134*)

Nesse exercício o objetivo era mudar o departamento do Tobi Barnett para 'Sales'. Então eu fiz um comando de Update mudando o departamento através do id do usuário. (*UPDATE employees SET department = 'Sales' WHERE userid = 89762*)

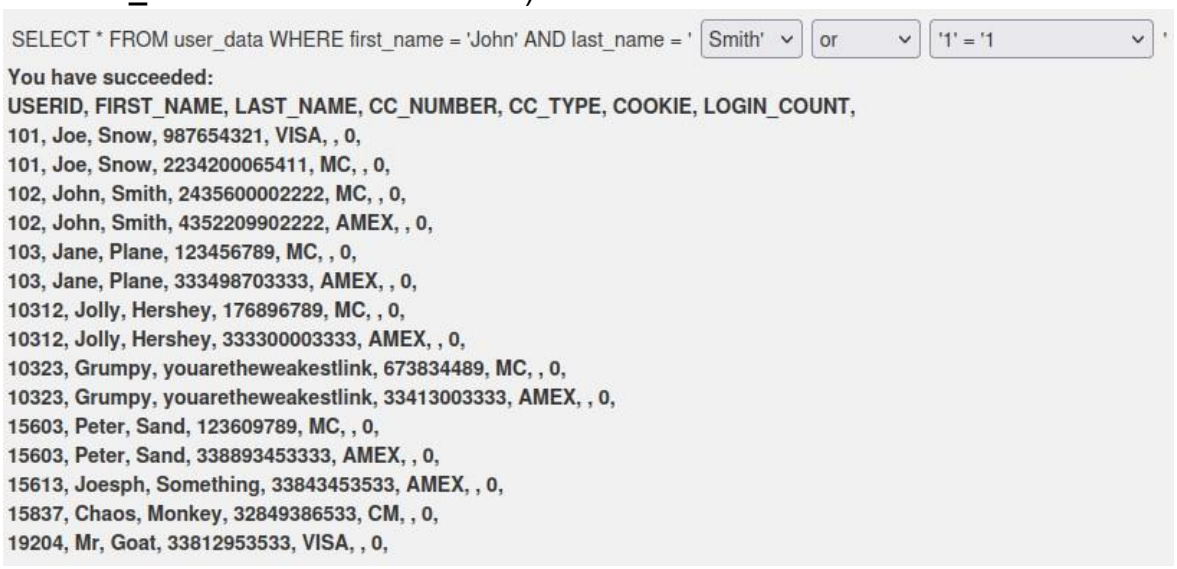


Nesse exercício o objetivo era modificar o esquema adicionado a coluna “phone” (varchar(20)) na tabela “employees”. Então eu fiz um comando de Alter Table para adicionar a nova coluna. (*ALTER TABLE employees ADD phone varchar(20)*)

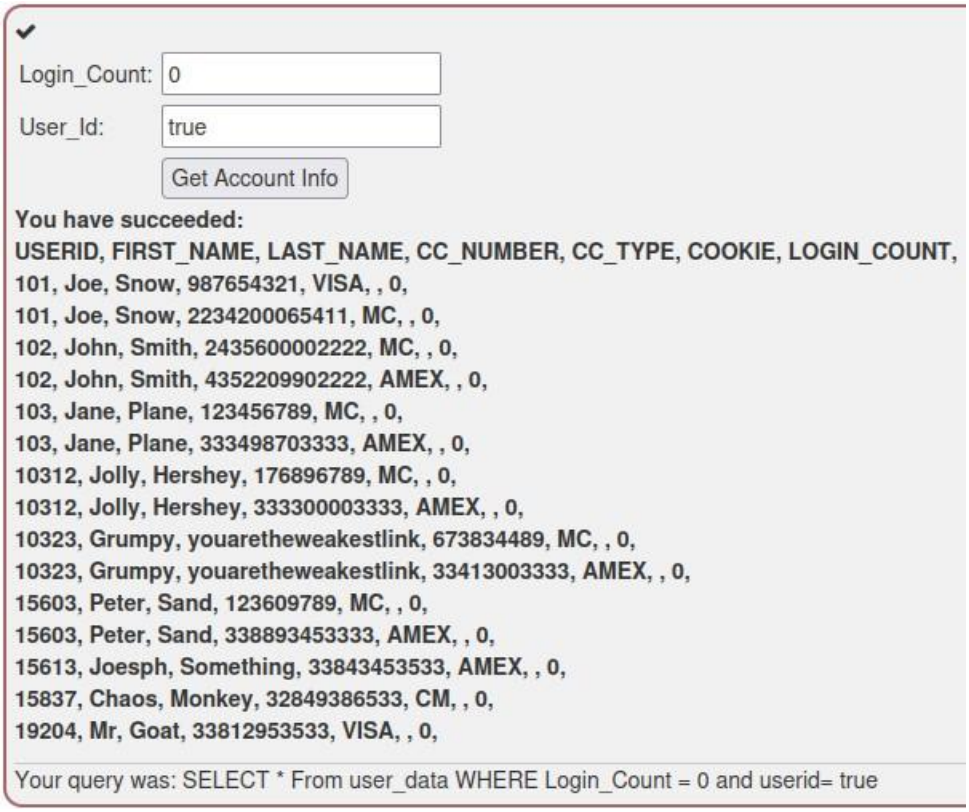
Nesse exercício o objetivo era dar os direitos de acesso da tabela ‘grant_rights’ para o usuário ‘unauthorized_user’. Então eu fiz um comando Grant permitindo diferentes comando ao usuário. (*GRANT SELECT, INSERT, UPDATE, DELETE ON grant_rights TO unauthorized_user*)



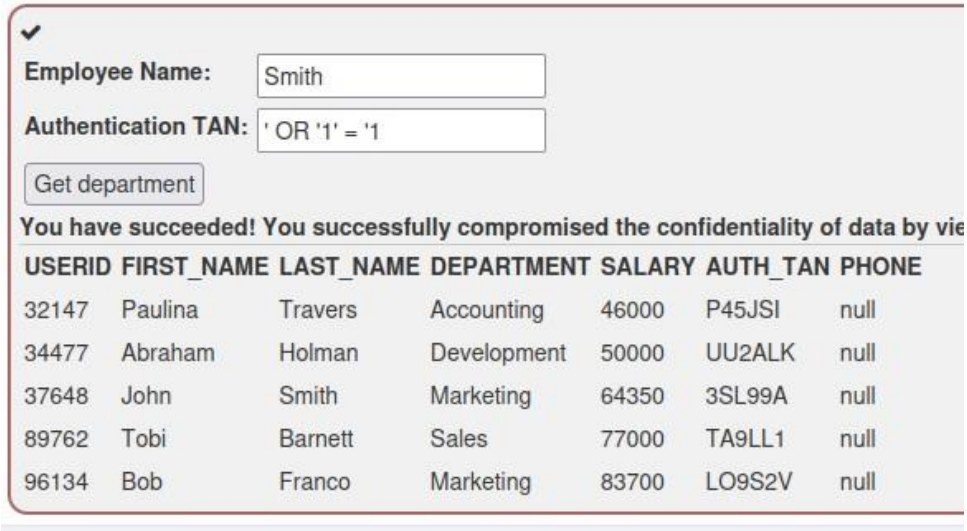
Nesse exercício o objetivo era usar o formulário abaixo para coletar todos os usuários da tabela usuário. Então eu selecionei as opções corretas para resultar na seguinte consulta (*SELECT * FROM user_data WHERE first_name = ‘John’ AND last_name = ‘Smith’ OR ‘1’ = ‘1’*)



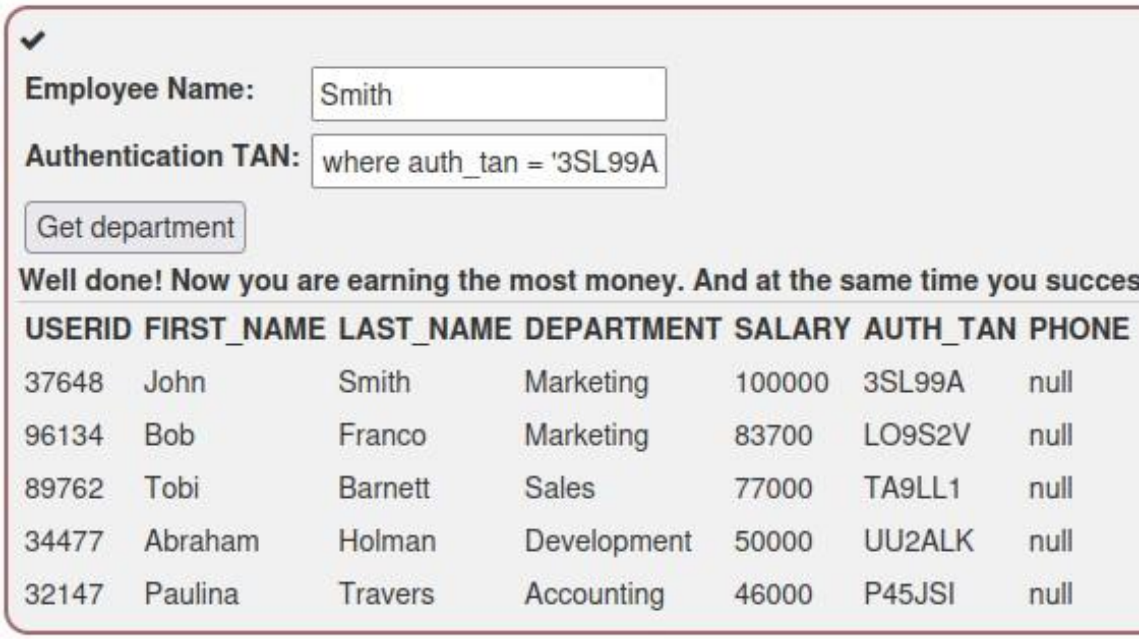
Nesse exercício o objetivo era usar os dois campos abaixo, tentando coletar todos os dados da tabela usuário. Então eu escrevi as duas opções corretas para resultar na seguinte consulta (*SELECT * FROM user_data WHERE Login_Count = 0 and userid = true*)



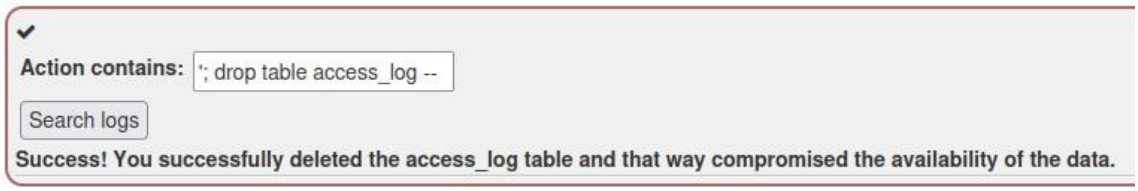
Nesse exercício o objetivo era usar o formulário abaixo para tentar coletar todos os dados de funcionário da tabela funcionários. Então eu escrevi as duas opções corretas para resultar na seguinte consulta (*"SELECT * FROM employees WHERE last_name = ' " + Smith + " ' AND auth_tan = '" + ‘ OR ‘1’ = ’1 + " ' "*)



Nesse exercício o objetivo era mudar o próprio salário para ser o mais bem pago. Então eu escrevi as duas opções corretas para resulta na seguinte consulta (*SELECT * FROM employees WHERE last_name = 'Smith' AND auth_tan = ' '; UPDATE employees SET salary = 100000 WHERE auth_tan = '3SL99A'*)



Nesse último exercício o objetivo era deletar a tabela 'access_log' para apagar os rastros do que foi feito. Então eu escrevi a opção correta para resultar na seguinte consulta (*SELECT * FROM access_log WHERE action LIKE % '; DROP TABLE access_log --%*)



7. Referência Bibliográficas

Open Application Security Project, Disponível em: https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

OWASP. SQL Injection. Disponível em: https://owasp.org/wwwcommunity/attacks/SQL_Injection

PortSwigger. SQL Injection. Disponível em: https://portswigger.net/websecurity/sql-injection

WebGoat. GitHub. Disponível em: https://github.com/WebGoat/WebGoat

OWASP. Projeto OWASP. Disponível em: https://owasp.org/www-community/