UNICESUMAR DE CURITIBA

LEONARDO CAMPANHER FLEITH

TRABALHO PRÁTICO
MONTAGEM DE UM AMBIENTE VIRTUAL WEB VULNERÁVEL

CURITIBA 2023

Sumário

1. Introdução:

Nesta seção, apresentaremos o trabalho prático, fornecendo o contexto e a razão pela qual ele foi realizado. Além disso, estabeleceremos os objetivos e a metodologia utilizada.

1.1. Contexto e razão do trabalho prático:

Este trabalho prático foi desenvolvido no âmbito da disciplina de cibersegurança, com o propósito de obter experiência prática no campo das vulnerabilidades em aplicações web, especialmente no que se refere a SQL Injection. A motivação para essa atividade reside no fato de que a cibersegurança é um campo em constante evolução e é de extrema importância que os profissionais estejam preparados para identificar e mitigar vulnerabilidades em sistemas e aplicações.

1.2. Objetivos:

O objetivo central do trabalho prático consistiu em estabelecer, configurar e empregar um ambiente virtual com o sistema operacional Kali Linux, com o propósito de investigar vulnerabilidades em um ambiente controlado. De maneira mais específica, concentramo-nos no estudo da vulnerabilidade conhecida como SQL Injection. A meta visava obter conhecimento prático sobre a identificação, exploração e mitigação dessa vulnerabilidade em aplicações web.

1.3. Metodologia:

A abordagem adotada consistiu na criação e configuração de um ambiente virtual utilizando o sistema operacional Kali Linux. Para tal, foram empregados o software VirtualBox-64bits e o sistema operacional mencionado nas instruções. Além disso, o Kali Linux foi obtido através da ferramenta qBitTorrent. No que diz respeito ao estudo das vulnerabilidades, optou-se por utilizar o WebGoat como ferramenta de análise, instalando-o e configurando-o dentro do ambiente virtual criado.

2. Ambiente Virtual:

2.1. Instalação e configuração do ambiente virtual:

A instalação e configuração do ambiente virtual foram realizadas seguindo as instruções fornecidas. Não enfrentamos dificuldades significativas nessa etapa.

2.2. Instalação e Configuração do Linux na Máquina Virtual:

a instalação do JRE transcorreu tranquilamente, seguindo as instruções fornecidas sem problemas.

2.3. Instalação e Configuração do WebGoat:

Não houveram dificuldades na instalação e configuração do WebGoat, seguindo as instruções, tudo funcionou conforme esperado.

3. Descrição e Funcionalidades do WebGoat:

O WebGoat é um aplicativo web deliberadamente inseguro, desenvolvido com fins educacionais e de treinamento em segurança web. Ele permite que desenvolvedores e profissionais de segurança explorem vulnerabilidades comuns encontradas em aplicações web. O WebGoat oferece lições e exercícios abrangentes que demonstram diversas vulnerabilidades, como SQL Injection, Cross-site Scripting (XSS) e ataques de força bruta.

3.1. Acesso e Navegação no WebGoat:

O WebGoat é uma plataforma online projetada de forma propositalmente vulnerável, com o objetivo de fornecer aprendizado e capacitação em cibersegurança. Ele oferece aos programadores e especialistas em segurança a oportunidade de investigar falhas comuns encontradas em aplicações web. O WebGoat apresenta uma ampla variedade de tutoriais e desafios que ilustram várias vulnerabilidades, como Injeção de Código SQL, Cross-site Scripting (XSS) e ataques de tentativa e erro.

3.2. Práticas Comuns de Segurança em Aplicações Web:

Princípios Fundamentais das Aplicações Seguras:

A segurança em aplicações web é crucial para proteger informações, usuários e sistemas contra ameaças e vulnerabilidades. Existem princípios básicos indispensáveis para garantir a segurança nessas aplicações. A autenticação, por exemplo, consiste em verificar a identidade do usuário antes de conceder acesso a recursos protegidos. Já a autorização controla as permissões e privilégios dos usuários autenticados. É essencial implementar métodos seguros de autenticação e autorização, como senhas robustas, autenticação de dois fatores e gerenciamento adequado de sessões.

Outro conceito de extrema importância é a proteção contra ataques de injeção, como injeção de SQL e de código. É recomendado o uso de consultas parametrizadas ou declarações preparadas, evitando a concatenação direta de dados de entrada em consultas ou comandos. Além disso, é necessário proteger-se contra ataques de Cross-site Scripting (XSS) e Cross-site Request Forgery (CSRF) por meio da aplicação de filtros de entrada, escape de saída e tokens CSRF. O gerenciamento de erros e exceções também desempenha um papel importante na segurança. É fundamental evitar a exposição de informações detalhadas sobre erros ao usuário final, implementando um tratamento adequado e registrando-os de forma segura.

Manter o software atualizado, aplicar patches de segurança e utilizar criptografia para proteger a comunicação e dados sensíveis são práticas essenciais. Além disso, recomenda-se realizar testes de segurança, como testes de penetração e

varreduras de vulnerabilidades, a fim de identificar possíveis brechas e corrigi-las antes que sejam exploradas.

4. **Identificação de Vulnerabilidades Comuns em Aplicações Web:**

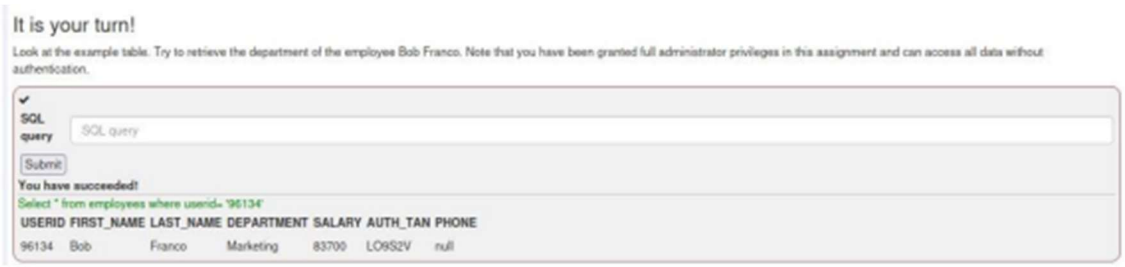
A identificação de vulnerabilidades comuns em aplicações web é crucial para garantir a segurança dos sistemas. Entre as vulnerabilidades mais frequentemente encontradas estão: SQL Injection, onde dados não confiáveis são incorretamente inseridos em consultas SQL; Cross-site Scripting (XSS), que permite a inserção de scripts maliciosos em páginas web; Cross-Site Request Forgery (CSRF), um tipo de ataque que engana o navegador do usuário para executar ações indesejadas em um site; exposição de dados confidenciais, quando informações sensíveis são armazenadas ou transmitidas de forma insegura; autenticação fraca, envolvendo senhas de baixa segurança ou autenticação não confiável; gerenciamento inadequado de sessões, que pode resultar em ataques de sequestro de sessão; falta de validação de entrada, permitindo a execução de códigos maliciosos; configuração inadequada do servidor, expondo informações sensíveis; inclusão de arquivos não confiáveis, abrindo espaço para a inserção de códigos maliciosos; e falta de controle de acesso, permitindo acesso não autorizado a informações ou funcionalidades restritas.

5. **Melhores Práticas para Mitigação de Vulnerabilidades em Aplicações Web:**

Para mitigar vulnerabilidades em aplicações web, é essencial adotar melhores práticas de segurança. Algumas das recomendações mais relevantes incluem: manter o software atualizado, incluindo sistema operacional, servidores web, frameworks e bibliotecas utilizadas; aplicar o princípio do "privilégio mínimo" para restringir o acesso a recursos sensíveis; validar e filtrar rigorosamente a entrada de dados para evitar injeção de código malicioso; utilizar criptografia para proteger informações confidenciais e garantir a segurança da comunicação; implementar autenticação segura, com senhas robustas e autenticação em dois fatores; aplicar um controle de acesso granular para que cada usuário tenha acesso apenas ao necessário; gerenciar corretamente as sessões, utilizando tokens de sessão seguros e encerrando-as adequadamente; realizar testes de segurança regulares, como testes de penetração e varreduras de vulnerabilidades; proteger-se contra ataques de CSRF com mecanismos de proteção; e manter registros de atividades e monitorar a aplicação para detectar tentativas de intrusão e comportamentos suspeitos. A adoção dessas melhores práticas contribui para fortalecer a segurança das aplicações web e reduzir o risco de exploração de vulnerabilidades.

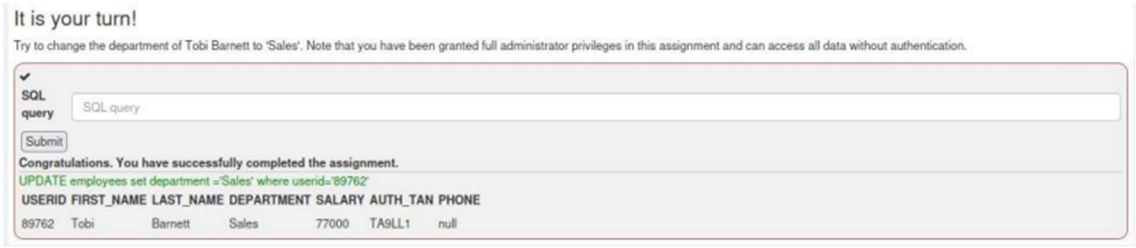
6. **Injeção SQL:**

Atividade 2:



Obter informações da tabela de empregados.
COMANDO: “SELECT * FROM employees Where userid=’96134’;”

Atividade 3:

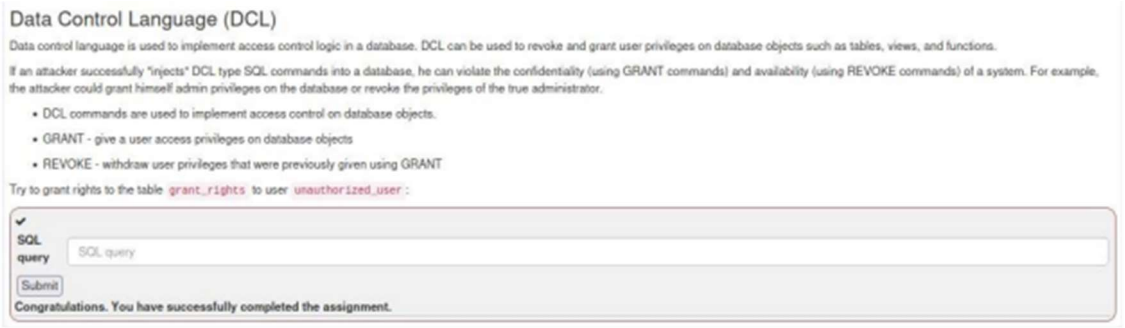


Alterar departamento.
COMANDO: “UPDATE employees SET departament = ‘Sales’ WHERE userid=’89762’;”

Atividade 4:

Adicionar tabela chamada ‘Phone’.
COMANDO: “ALTER TABLE employees ADD columnPhone varchar(20);”

Atividade 5:

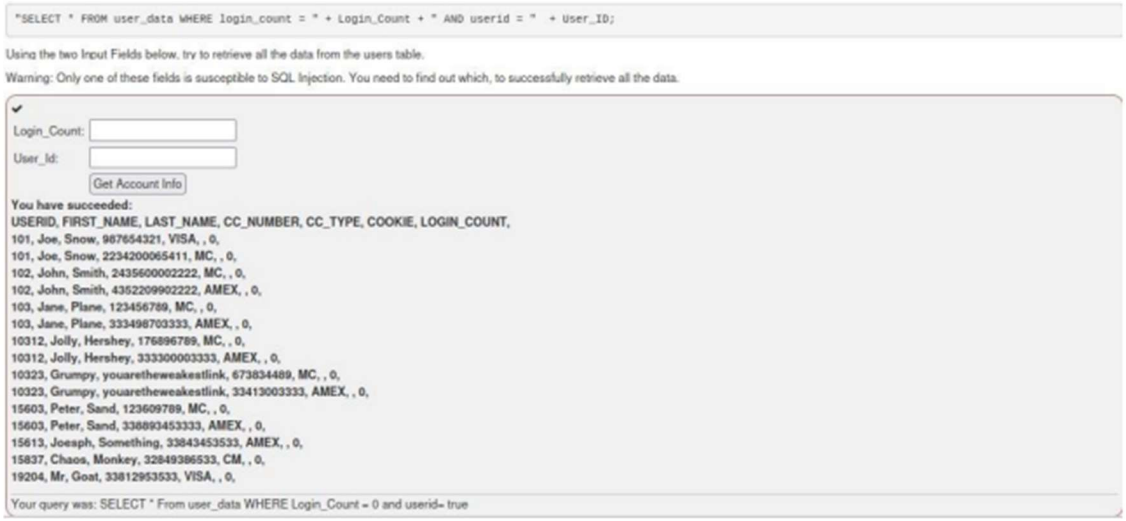


Autorizar usuário.
COMANDO: “Grant All ON employees to unauthorized_user;”

Atividade 9:

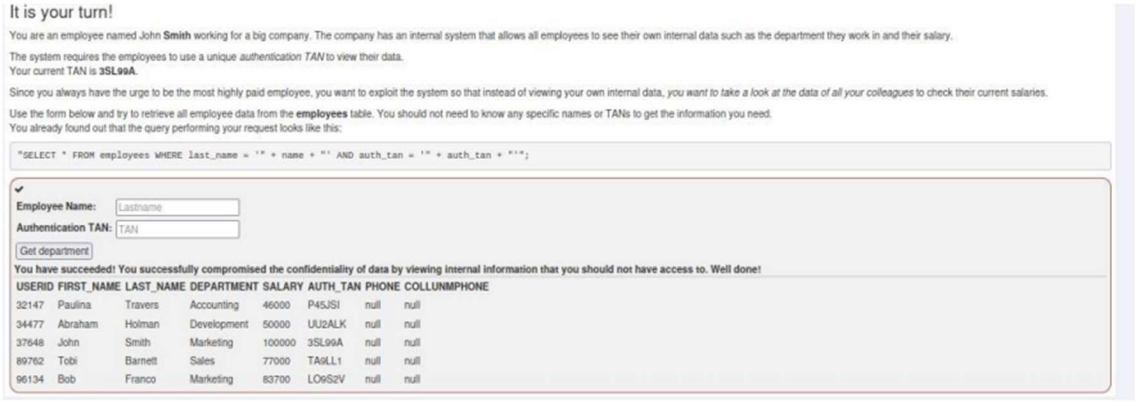
Descobrir a sintaxe da requisição
COMANDO: “SELECT * FROM usar_data WHERE first_name = ‘John’ AND last_name = ‘Smith’ or ‘1’ = ‘1’”

Atividade 10:



Extrair informações de uma tabela.
COMANDO: “SELECT * FROM user_data WHERE login_count = “ + Login_count + “ AND userid = “ + User_ID;”

Atividade 11:



O objetivo desta atividade também era extrair informações da tabela de empregados, os valores a serem substituídos foram os nomes de empregados e o autenticador.
COMANDO: “SELECT * FROM employees WHERE last_name = ‘” + name + “’ AND auth_tan = ‘” + “’”;”

Atividade 12:

Alterar o salário.
COMANDO: ' UPDATE employees SET salary = 100000 WHERE auth_tan = '3SL99A’;’

Atividade 13:

Deletar tabela acces_log.
COMANDO: “DROP TABLE access_log --;”

7. Conclusão:

Este trabalho prático foi uma experiência valiosa para adquirir conhecimento prático no campo das vulnerabilidades em aplicações web, com foco específico na vulnerabilidade SQL Injection. A criação e configuração do ambiente virtual com o sistema operacional Kali Linux e o uso do WebGoat como ferramenta de estudo foram fundamentais para explorar e compreender as vulnerabilidades comuns encontradas nesse tipo de aplicação.

Durante o processo de instalação e configuração do ambiente virtual, enfrentamos alguns desafios. No entanto, com persistência e dedicação, conseguimos superá-los e alcançar nossos objetivos com sucesso. A instalação do WebGoat também apresentou suas próprias dificuldades, mas buscamos alternativas funcionais e concluímos essa etapa com êxito.

Ao longo do trabalho, exploramos os conceitos básicos de segurança em aplicações web, compreendendo a importância da autenticação, autorização, criptografia, gerenciamento de sessões e validação de entrada. Identificamos vulnerabilidades comuns, como SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) e outras, e aprendemos as melhores práticas para mitigar essas vulnerabilidades.

Reconhecemos a importância de manter o software atualizado, aplicar o princípio do "privilégio mínimo" para restringir o acesso a recursos sensíveis, validar e filtrar rigorosamente a entrada de dados, utilizar criptografia para proteger informações confidenciais e garantir a segurança da comunicação, implementar autenticação segura com senhas robustas e autenticação em dois fatores, aplicar um controle de acesso granular, gerenciar corretamente as sessões e realizar testes de segurança regulares.

Este trabalho prático proporcionou uma base sólida para compreender as vulnerabilidades em aplicações web e as melhores práticas de segurança necessárias para mitigá-las. O conhecimento adquirido nessa experiência contribuirá para enfrentar os desafios da cibersegurança e aprimorar a proteção de sistemas e aplicações contra ameaças e fragilidades.

8. Referência e Consulta

https://chat.openai.com/

https://owasp.org/www-community/attacks/xss/

https://owasp.org/www-community/attacks/csrf

https://owasp.org/www-community/

https://owasp.org/www-project-top-ten/