Pentest com SQLMap

Juan Felipe Serafim dos Santos

[1] Centro de Informática
Universidade Federal de Pernambuco (UFPE) – Recife, PE – Brazil

jfss@cin.ufpe.br

Resumo. *Pentest, ou teste de penetração, é uma prática de segurança da informação que consiste em simular ataques controlados a sistemas computacionais, redes e aplicações, com o objetivo de identificar vulnerabilidades. O principal objetivo dessa atividade é avaliar o nível de segurança de um ambiente, descobrir falhas técnicas ou lógicas e fornecer recomendações para corrigi-las antes que sejam exploradas em ataques reais.*

1. Objetivo

Para a proposta da atividade, necessitaremos de um ambiente vulnerável a SQL Injection para a utilização do SQLMap. Uma maneira simples de se conseguir isso é utilizando o *Damn Vulnerable Web Application* (DVWA). O DVWA é uma aplicação web propositalmente vulnerável, desenvolvida para fins educacionais e de testes em segurança da informação. E o SQLMap é uma ferramenta de teste de penetração de código aberto que automatiza o processo de detecção e exploração de falhas de injeção de SQL e o controle de servidores de banco de dados.

2. Como a ferramenta funciona

Após toda configuração inicial necessária para o DVWA funcionar devidamente, sua tela inicial é da seguinte forma:

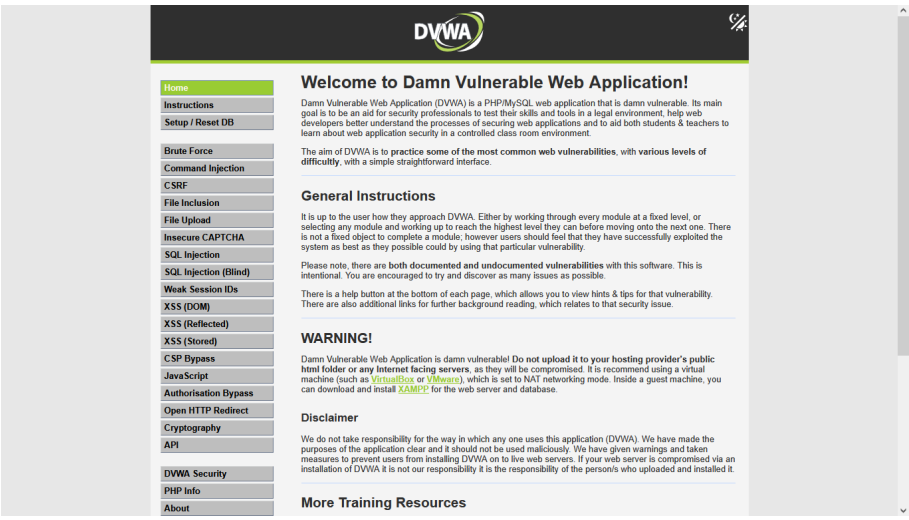

Figura 1. Tela inicial do DVWA

Para demonstrar o ataque SQL Injection, será configurado o nível de segurança para o **menor possível**.

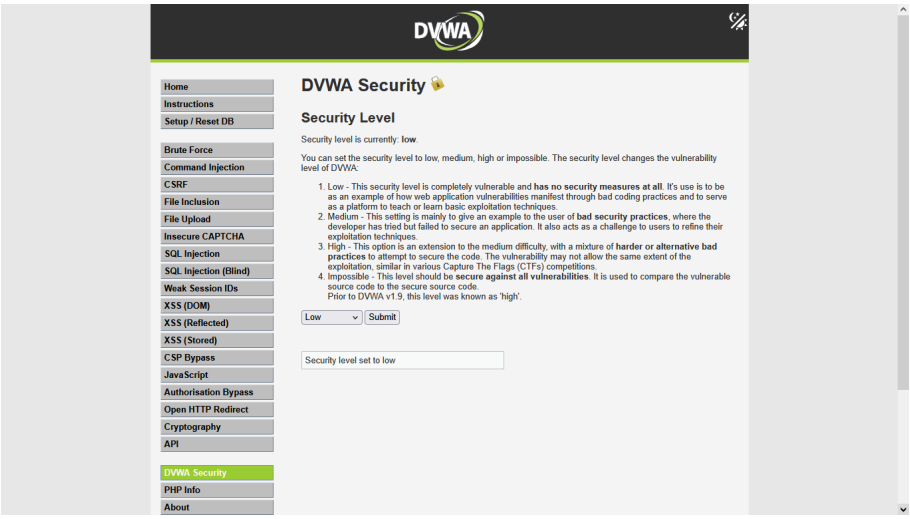

Figura 2. Configuração do nível de segurança

Para o SQLMap, temos tal tela inicial:

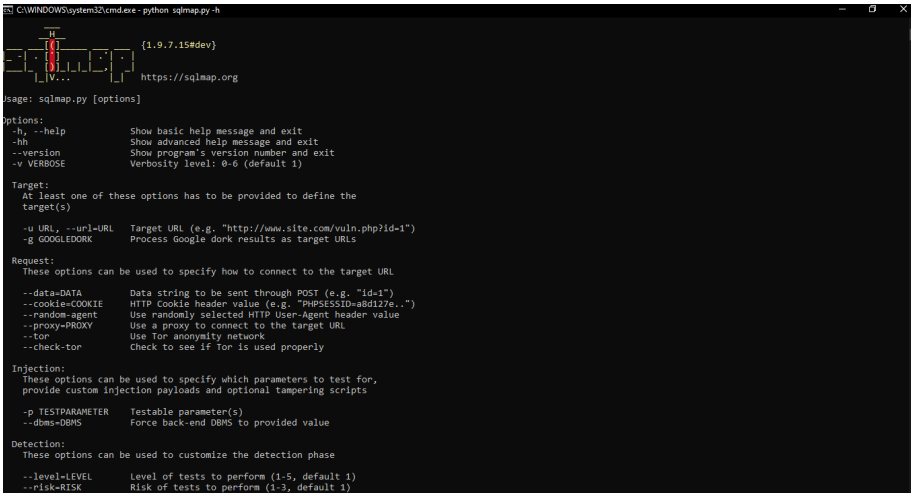

Figura 3. SQLMap

Esse é somente um trecho de todos os parâmetros que podem vir a ser utilizados. São muitos. Portanto, irei detalhar somente o que eu for utilizar. Então:

- -u URL, –url=URL URL Alvo (e.g. ”http://www.site.com/vuln.php?id=1”)
- –cookie=COOKIE HTTP Cookie valor cabeçalho (e.g. ”PHPSESSID=a8d127e..”)
- –dbms=DBMS Forçar a análise para o SGBD back-end

Após a execução dos parâmetros **-u “http://localhost/dvwa/vulnerabilities/sqli/?id=&Submit=Submit” –cookie=“PHPSESSID=8b9ns5g1nefddqddhkvaetfgjv;security=low” –dbms=mysql**

Obs: esse valor de cookie depende do ambiente de execução, possivelmente será diferente em outra tentativa.

O resultado da execução do SQLMap retornou tais *payloads* vulneráveis:

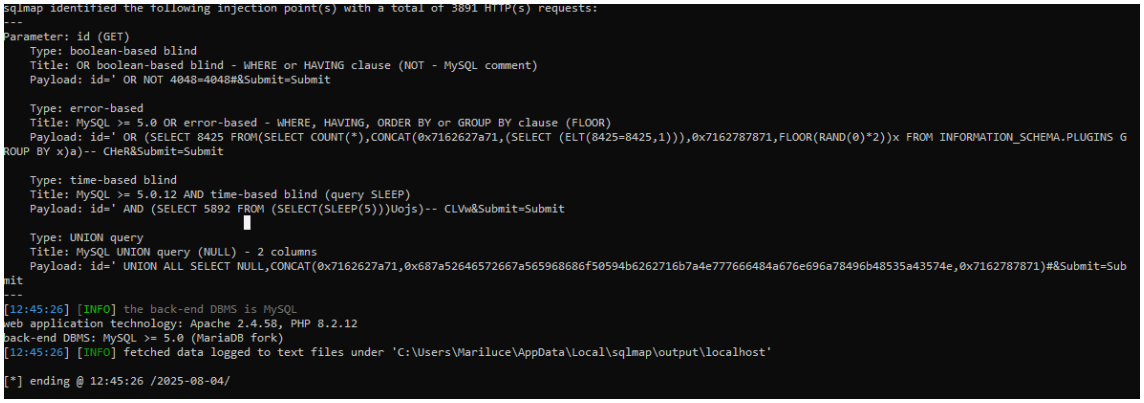

Figura 4. Resultado do SQLMap com os payloads maliciosos

3. Análise da vulnerabilidade

A vulnerabilidade em questão é o SQL Injection, que é uma falha de segurança em aplicações web que permite a um atacante interferir nas consultas SQL feitas ao banco de dados. Essa falha ocorre quando uma aplicação aceita entradas fornecidas pelo usuário **sem validar ou higienizar** adequadamente os dados antes de usá-los em comandos SQL.

4. Sugestão de defesa

Um bom tratamento de como as consultas serão submetidas ao servidos, com o uso das chamadas consultas parametrizadas (prepared statements), que garantem a separação entre o código SQL e os dados fornecidos pelo usuário, validação rigorosa dos dados de entrada, especialmente em campos como formulários, URLs e parâmetros de requisição e, principalmente, sanitizar os dados.

Referências

Bellucci, D. (2006). sqlmap® automatic sql injection and database takeover tool. https://sqlmap.org/.

Dewhurst, R. (2008). Damn vulnerable web application (dvwa). https://github.com/digininja/DVWA.

Galdino, G. (2022). Eficiência e segurança com consultas parametrizadas: prevenindo sql injection. https://dev.to/gabogaldino/eficiencia-e-seguranca-com-consultas-parametrizadas-prevenindo-sql-injection-5bln.