Segurança
Experimento N°04
Vulnerabilidade de Aplicações Web

Disciplina	**Segurança em Redes de Computadores**
Professor:	**Rafael Timóteo de Sousa Júnior**
Monitores:	**Valério Martins (valerioaymoremartins@gmail.com)** **Mariana Stieljes**

1. Introdução:

O **OWASP** (Open Web Application Security Project) é uma comunidade online que produz artigos gratuitos, metodologias documentações, ferramentas e tecnologias na área de segurança de aplicações web.

Uma de suas aplicações mais famosas é o **OWASP Top Ten** (desde 2003) que é atualizado regularmente. Esta publicação tem o intuito trazer à conhecimento público os maiores riscos que as empresas no mercado têm enfrentado em termos da segurança de suas aplicações. Muitos padrões, livros e organizações utilizam o projeto **Top Ten** como referência. A última versão do documento indica que as dez maiores (comentadas em sala de aula) explorações críticas da segurança de aplicações web atualmente são:

- **A1 Injection**
- **A2 Broken Authentication and Session Management**
- **A3 Cross-Site Scripting (XSS)**
- **A4 Insecure Direct Object References**
- **A5 Security Misconfiguration**
- **A6 Sensitive Data Exposure**
- **A7 Missing Function Level Access Control**
- **A8 Cross-Site Request Forgery (CSRF)**
- **A9 Using Components with Known Vulnerabilities**
- **A10 Unvalidated Redirects and Forwards**

É de interesse não só de atacantes mal-intencionados, mas também dos mantenedores e da equipe que desenvolve qualquer projeto, identificar falhas de segurança que podem comprometer completamente ou parcialmente sua aplicação e/ou dados de seus clientes.

Pensando nisso, a comunidade de Pentesters e profissionais de segurança de aplicações desenvolveu algumas ferramentas que automatizam a busca pelas falhas mais comuns indicadas pela OWASP em qualquer aplicação web. Apesar do trabalho de identificação e exploração ser um trabalho que requer bastante conhecimentos específicos e criatividade de um atacante, é possível, de maneira confiável, automatizar a maior parte do trabalho de identificação.

O objetivo deste experimento é utilizar duas dessas ferramentas de automação (**WPScan** e **OWASP ZAP)** para identificar falhas de segurança críticas em duas aplicações (**Juice Shop*** e um **Wordpress** propositalmente vulnerável) e, afim de demonstrar o processo de *exploit* de um atacante, utilizaremos um script do **Metasploit** para comprometer o Wordpress.

***ou qualquer outra aplicação web da página (à critério do aluno):**
https://www.owasp.org/index.php/OWASP_Vulnerable_Web_Applications_Directory_Project [1]

2. Ambiente e Necessidades Preliminares

- *Para **sala de aula (ambiente preparado pelo professor, portanto não há necessidade de configuração por parte do aluno):***
    - ✓ Configuração de ambiente inseguro (Wordpress + PHP-FPM + MySQL todos *outdated*):
        - ❖ Wordpress 3.0: Lançado em Junho de 2010
        - ❖ PHP5.0-FPM: Lançado em Julho de 2004
        - ❖ MySQL 5.5: Lançado em Dezembro de 2010

- Para **casa (requer configuração por parte do aluno):**
    - ✓ Configuração de uma das aplicações [1] em sua máquina local. Para exemplificação será configurado o Juice Shop (link para download: https://github.com/bkimminich/juice-shop).

Segundo a documentação (no próprio repositório), o **Juice Shop** possui mais de 4 modos de instalação.

O exemplo abaixo ilustra a instalação via Docker (recomendado).

(A) Instalação do Docker:

1 Adicionar chave do repositório:
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

2 Adicionar repositório:
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

3 Atualizar lista de pacotes:
$ sudo apt-get update

4 Instalar docker:
$ sudo apt-get install -y docker-ce

5 Verificar se serviço está rodando:
$ sudo systemctl status docker

(B) Instalação do Juice Box:

1 Baixar imagem do DockerHub:
$ docker pull bkimminich/juice-shop

2 Rodar container:
$ docker run --rm -p 3000:3000 bkimminich/juice-shop

3 Acessar no browser **http://localhost:3000**

3. Experimento

3.1 Parte de Sala

a) Scan de vulnerabilidades do Wordpress via OWASP ZAP e WPScan
b) Exploração de vulnerabilidade utilizando Metasploit

3.2 Parte de Casa

a) Subir serviço vulnerável [1] ou [2]
b) Realizar scan do serviço via OWASP ZAP e descrever vulnerabilidades encontradas
c) Explorar manualmente **duas** das vulnerabilidades encontradas e descrever processo de ataque e consequências.*

* Todas as aplicações listadas possuem solução na internet (pesquisar por exemplo por "Juice Shop solutions").

* A ideia é que se implemente manualmente dois dos exploits da aplicação explicando o processo e sua consequência para a aplicação.