Anterior • Trilha I • Próximo comentar

Lição 3 - Construindo um Portscanner em Java

- **Objetivo(s):** Oferecer informações de como utilizar os sockets na criar um Port Scanner.
- **Direitos autorais e licença:** Veja notas de direitos autorais e licença no final da lição.

Conteúdo

3.1 - Criação do *PortScanner*

Agora que já vimos como criar e fechar *sockets* usando o TCP vamos desenvolver uma aplicação simples, mas que pode ser útil em muitos casos: um *Port Scanner*.

Um ***Port Scanner***, ou *scanner* de porta, é um aplicativo que tem como objetivo testar as portas lógicas de determinado *host* remoto, buscando identificar quais estão fechadas e quais estão esperando conexão. É possível explicitar o limite de portas que o aplicativo irá escanear, por ex: 1 a 1000. Geralmente esses aplicativos são usados por pessoas mal intencionadas para identificar portas abertas e planejar invasões, mas também podem ser usados por empresas de segurança para análise de vulnerabilidades.

Podemos construir nosso aplicativo usando a API de *sockets*. Basta tentar criar uma conexão com o *host* escaneado, usando as portas analisadas. Se a conexão for bem sucedida o *host* está escutando naquela porta. Se o host não tiver esperando conexões nesta porta uma exceção do tipo *java.io.IOException* é gerada.

Dessa forma, podemos testar a porta 0 usando o código abaixo:

```
import java.io.IOException;
import java.net.Socket;

public class PScanner {
        public static void main(String[] args) {
                String host = "localhost";
                try {
                        Socket s = new Socket(host, 0);
                        System.out.println("Servidor está esperando conexão na porta " + porta + " de " + host);
                        s.close();
                } catch (IOException ex) {
                        System.out.println("Servidor não está esperando conexão na porta " + porta + " de " + host);
                }
        }
}
```

Baixe o código-fonte acima neste link: http://wiki.marceloakira.com/pub/GrupoJava/ConstruindoUmPortscannerEmJava/PScanner.java

Como o objetivo é testar todas as portas, podemos colocar o código acima dentro de um laço **for** e usar a variável do laço para tentar criar o *socket*. Desse modo, iremos testar todas as portas do *host*. Veja o código abaixo:

```
import java.io.IOException;
import java.net.Socket;

public class PScanner {

        public static void main(String[] args) {
                String host = "localhost";

                for (int porta = 0; porta < 65536; porta++) {
                        try {
                                Socket s = new Socket(host, porta);
                                System.out.println("Servidor está esperando conexão na porta " + porta + " de " + host);
                                s.close();
                        } catch (IOException ex) {
                                System.out.println("Servidor não está esperando conexão na porta " + porta + " de " + host);
                        }
                }
        }
}
```

Baixe o código-fonte acima neste link: http://wiki.marceloakira.com/pub/GrupoJava/ConstruindoUmPortscannerEmJava/PScanner.java

Para testar outro host basta trocar "localhost" pelo nome do host em questão.

3.2 - Direitos autorais e licença

Anterior • Trilha I • Próximo

3.3 - Comentários

Adicionar