Parte 2 – Exercício 2b)

Neste caso, a classe HttpURLConnection é determinada como Source, pois é feito um inputStream da mesma, e este inputStream irá buscar dados a um URL fornecido na aplicação, e deste URL virão dados que passarão para o InputStream que por sua vez passará esses dados a algum lado para serem processados. A razão de ser uma Source, ou fonte é exatamente porque está a ir buscar os dados para dentro da aplicação.

No caso do FileOutputStream ser classificado como Sink, é relativamente óbvio, tendo em conta que esta classe serve exclusivamente para escrever dados para fora do programa. Tendo isso em conta a sua classificação como Sink parece correta pois está a deixar os dados sair para outro sitío, que neste caso será provavelmente um ficheiro.

Parte 2 – Exercício 2c)

Para fazer download dos ficheiros para o dispositivo será necessário um Source que vai buscar os APK onde quer que estejam, tendo em conta que é feito o Download dos mesmos, e posteriormente um Sink que tratará de os instalar no dispositivo.

Nesse caso é pedido como é feito o download, e neste caso é utilizado um dos exemplos apresentados no exercício anterior nomeadamente o da figura seguinte.

```
<Source Statement="$r12 = virtualinvoke $r5.&lt;java.net.HttpURLConnection: java.
io.InputStream getInputStream()&gt;()" Method="&lt;com.exchange.Public.Downloadin
gService$a: void a()&gt;">
  ...
</Source>
```

Podemos dizer com certeza que o download ocorre aqui, pois este método permite a chamada a um URL de escolha de quem fez o programa, e acaba por retornar um InputStream. Esta classe trará consigo todos os bytes que vierem do URL que será potencialmente suspeito pois não há verificação de tal coisa. Desta forma o atacante poderá colocar um URL que ele criou com APKs corruptos, que depois serão acedidos por aqui, e a partir daí provavelmente instalados no dispositivo.