Universidade do Minho - Dep.to Informática
1º Semestre, 2018/2019
Gestão de Redes

Trabalho Prático Nº3

Servidor de criação e monitorização de containers com interface SNMP

Objectivos:

- Consolidação da utilização prática do modelo de gestão preconizado pelo *Internet-standard Network Management Framework* (INMF), dando especial relevo ao *Simple Network Management Protocol* (SNMP) e às *Management Information Bases* (MIBs).
- Utilização de APIs SNMP para construção de ferramentas de gestão (agentes e gestores).
- Investigação da aplicação do SNMP em sistemas de gestão nos mais variados ramos da engenharia aplicacional.

Observações:

- O trabalho deverá ser realizado em cerca de 70 horas efetivas de trabalho.

Requisitos:

- Sistema com um agente SNMPv2c instalado (preferencialmente o NET-SNMP) e pacote de desenvolvimento numa linguagem de programação que disponibilize APIs para construção de gestor e agente SNMPv2c (como por exemplo o SNMP4J).

AVISOS:

- Não serão tolerados atropelos aos direitos de autor de qualquer tipo de *software*...

Bibliografia específica e material de apoio

Material de apoio:

- Manuais do *ucd-snmp* e *scottty*
- MIBs em `/usr/share/snmp/mibs` e `/aplicacoes/MIBs`
- Recurso `http://net-snmp.sourceforge.net/wiki/index.php/Tutorials/`
- Recurso `http://www.simpleweb.org/`
- Recurso `http://www.snmplinks.org/`
- Recurso `http://www.agentpp.com/`

Bibliografia:

- M. Rose, *The Simple Book*, Second Edition, Prentice Hall, 1996.
- W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison-Wesley, 2000.
- D. Mauro, K. Schmidt, *Essential SNMP*, O'Reilly, 2001.
- Ver outros recursos fornecidos pela equipa docente.

Servidor de criação e monitorização de containers com interface SNMP

O objetivo principal deste trabalho é o desenvolvimento de um agente SNMP que seja um servidor de criação e gestão de containers. Este tipo de serviço remoto através da internet estaria assim disponível a outros sistemas para criar e gerir containers, que possam depois ser utilizados por outros utilizadores. Já existem serviços parecidos, fornecidos pela Amazon ou Google, mas este deverá ser acessível usando SNMP. O serviço criado deverá permitido a um administrador criar containers a partir de um conjunto de imagens pré-determinadas e consultar o estado dos containers existentes usando SNMP.

O primeiro passo do trabalho deve ser a definição dos requisitos funcionais, isto é, que tipo de resultados são esperados tendo em conta as possíveis parametrizações que o gestor pode fazer. Depois deve definir-se uma MIB com os grupos de objetos com a semântica e sintaxe adequadas à correta abstração dos requisitos funcionais pré-estabelecidos. Por fim, deve construir-se e testar-se o *software* do agente que implemente o serviço num agente SNMPv2c. Este *software* pode ser desenvolvido na linguagem e ambiente de programação que achar mais adequados.

Requisitos Funcionais Genéricos

Quando executado, o agente SNMP deve consultar um ficheiro de configuração contendo, no mínimo, os seguintes parâmetros de inicialização, um por linha, nesta ordem, todos obrigatórios:

- Porta UDP de atendimento de pedidos;
- Nome de comunidade SNMPv2c;

Deverá consultar também um ficheiro que contenha a lista de imagens (que serão usadas para criação dos containers) suportadas pelo agente.

Em conjunto com o ficheiro deste enunciado são disponibilizados dois ficheiros adicionais, um com um exemplo de configuração e outro com um conjunto de imagens que poderão ser utilizadas.

Como argumentos do programa (na linha de comandos) deverão ser os ficheiros de configuração e da lista de imagens.

Exemplo do comando para executar o agente:

```
containership-agent containership-conf.txt containership-images.txt
```

Containership MIB

A *Containership* MIB, a implementar no agente SNMP, deve conter quatro grupos:

- `containershipParam(1)` – grupo com objetos que representam um container novo a ser criado. Deverá conter pelo menos 3 objetos, o primeiro deverá indicar o índice da imagem a ser usada pelo container e o segundo qual o seu nome. O objeto final deverá ser apenas uma *flag* a indicar que este deverá ser criado. Todos os objetos deste grupo deverão ser *read/write*. Após a ativação *flag* (sinalizando a criação do *container*), deverá ser efetuado um *reset* a todos os objetos deste grupo e uma linha nova na tabela `containershipContainerTable` deverá ser criada com o novo *container*.

- `containershipImageTable(2)` – grupo com a tabela de imagens disponíveis; a tabela deve incluir pelo menos duas colunas, uma para o índice da entrada (que é chave da tabela) e o nome da imagem;

Index	Image
1	ubuntu:latest
2	postgres:8.4

- `containershipContainerTable(3)` – grupo com a tabela de *containers* que foram criados. Deverá ter pelo menos 4 colunas, o índice da entrada (chave da tabela), o nome do *container* correspondente, o índice da imagem que o originou, o seu estado (`up`, `down`, `removing`, `creating` e `changing`) e, finalmente, a percentagem de utilização do processador desse *container*. Todos os objetos deverão ser *read-only*, excetuando a coluna estado. Esta deverá poder permitir a escrita, sendo que o utilizador deverá poder escrever qual o estado para o qual deseje que o *container* passe. Caso o utilizador tenha pedido para mudar o estado, a coluna deverá passar para o estado `changing` até atingir o estado desejado.

Index	Name	Image	Status	Processor
1	Container1	1	up	95%
3	Container2	1	changing	65%

- `containershipStatus(4)` – grupo com objetos escalares que representem informações úteis e valores estatísticos de utilização do agente. No mínimo deverá conter a data do sistema quando o agente foi iniciado. Os alunos podem acrescentar outros objetos que julguem pertinentes.

FASE A – Definição *Containership* MIB

Para a primeira fase os alunos devem especificar uma *Containership* MIB e confirmar a sua sintaxe e semântica junto do docente antes de continuar para a Fase B.

FASE B – Implementação `containershipParam`

Para a segunda fase os alunos devem construir e testar um agente SNMP que leia o ficheiro de configuração e implemente apenas o grupo `containershipParam(1)` da *Containership* MIB (a lista de imagens iniciais indicado no ficheiro de configuração deve ser ignorado).

Os alunos devem confirmar com a equipa docente a correção do desenvolvimento do agente testando-o com as necessárias interações com comandos gestores do pacote NET-SNMP (ou equivalente).

FASE C – Implementação `containershipImageTable`

Na terceira fase os alunos devem acrescentar a implementação da `containershipImageTable (2)` da *Containership* MIB. A lista de imagens contida no ficheiro de configuração deverá ser carregada, introduzindo todos os valores lidos na tabela.

No final desta fase os alunos devem confirmar com o docente a correção do desenvolvimento do agente testando-o com as necessárias interações com comandos gestores do pacote NET-SNMP (ou equivalente).

FASE D – Implementação `containershipContainerTable`

Nesta fase, os alunos deverão fazer a implementação da `containershipContainerTable (3)`. A criação das linhas desta tabela deverá ser feita com base nos valores introduzidos em `containershipParam (1)`. Cada linha da tabela deverá ser criada inicialmente com a coluna `status` no estado *`creating`*, indicando que o referido *container* está a ser criado. Após a criação bem sucedida do *container*, esta *flag* deverá passar para o estado *`up`*.
Aquando da criação do container, os valores em `containershipParam (1)` deverão ser zerados.

FASE E – Implementação da mudança de estado

A operação de *mudança de estado* deve ser conseguida através do pedido duma operação SNMP-SET no agente ao objeto *status* na tabela `containershipContainerTable` com o valor de estado para o qual se deseja que o *container* mude. Este deverá passar imediatamente para o estado *`changing`* e, finalmente, para o estado em que ficou. Enquanto o valor estiver em *`changing`*, o agente não deverá aceitar novos pedidos de mudança de estado, a fim de evitar inconsistências graves.
No final desta fase os alunos devem confirmar com o docente a correção da operação de *mudança de estado* testando-a com as necessárias interações com comandos gestores do pacote NET-SNMP (ou equivalente).

FASE F – Implementação `containershipStatus`

Nesta fase os alunos devem acrescentar a implementação do grupo `containershipStatus(3)` da *Containership* MIB (apenas se todas as fases até à Fase D, inclusive, tiverem sido implementadas).

No mínimo, este grupo deverá conter a um objeto que armazene a data do sistema quando o agente foi iniciado e um contador. Adicionalmente, podem ser acrescentados outros objetos (escalares ou tabelas) que representem outros valores estatísticos interessantes, como por exemplo, o número de acessos à tabela, o número de aplicações gestoras que acederam ao agente, a média de acessos ao agente por hora ou por minuto, a largura de banda média de *output* do agente nos últimos Y minutos, o número de containers utilizados, quantas vezes cada imagem foi utilizada, etc.

No final desta fase os alunos devem confirmar com o docente a correção do desenvolvimento deste grupo no agente testando-o com as necessárias interações com comandos gestores do pacote NET-SNMP (ou equivalente).

FASE G – Implementação de medidas de segurança

Nesta fase os alunos devem discutir os problemas de segurança que um servidor deste género pode sofrer. Não se pretende aqui que abordem os problemas tecnológicos associados à utilização do SNMPv2c mas sim ameaças que advenham da utilização indevida do serviço, como por exemplo, um ataque com uma sequência de pedidos em número exagerado que obrigue o servidor, e eventualmente o sistema onde ele está a ser executado, a sentir dificuldades relevantes de funcionamento.

A melhor defesa para este tipo de ataques funcionais costuma ser integrar no agente um limitador ao número de pedidos que o agente poderá atender por segundo/minuto (para todos os pedidos e/ou por cada aplicação gestora) ou um limitador da largura de banda máxima disponível para output dos dados SNMP. Outra medida de segurança funcional é, por exemplo, permitir que todos os pedidos necessários para criar um *container* sejam feitas a partir do mesmo IP. Os alunos devem, no mínimo, tentar implementar uma medida funcional de segurança.

Esta fase deve ser abordada apenas se todas as funcionalidades básicas até à Fase C, inclusive, tiverem sido implementadas. No final desta fase os alunos devem confirmar com o docente a correção do desenvolvimento das funcionalidades de segurança funcional no agente testando-as com as necessárias interações com comandos gestores do pacote NET-SNMP (ou equivalente) ou até construindo uma pequena aplicação gestora que consiga fazer testes de *stress* funcional ao agente.

Relatório

Elabore o relatório do trabalho para ser entregue fisicamente durante a defesa do trabalho e também por *e-mail*. O relatório deve seguir o formato e a estrutura recomendada pela equipa docente (ver exemplo).

A primeira página do texto do relatório deve conter apenas, bem visível:

- Identificação do(s) aluno(s) (nome, número e fotografia).
- Identificação do trabalho em questão.
- Data da entrega.
- Nome do curso e da unidade curricular.

O texto do relatório deve conter uma secção a explicar a estrutura da ferramenta desenvolvida, um pequeno manual de utilização do agente e a estratégia seguida para a construção dos vários componentes das várias fases.

Além disso, deve conter também uma seção com os resultados da utilização efetiva do agente desenvolvido, incluindo *snapshots* dos resultados obtidos, em cada fase no sentido de validar a sua correção funcional.

Inclua o código da aplicação apenas no arquivo digital entregue por *e-mail*.

Excecionalmente, o texto do relatório pode conter uma secção extra denominada ***"Outras Considerações"*** com comentários genéricos sobre outras incidências que julgue importante comentar.