1 Teste automático remoto de servidores AS

O presente guia contém os procedimentos para execução remota de aplicações *user* na maquina 'tejo' para teste dos servidores AS desenvolvidos pelos alunos.

1.1 Introdução

No âmbito do projecto de comunicação usando a interface sockets oferece-se a possibilidade de testar as aplicações desenvolvidas pelos alunos em comunicação com aplicações que cumpram o protocolo especificado.
Tal é a finalidade do servidor concorrente AS em execução no porto 58011 da máquina 'tejo', o qual permite testar as aplicações *user* desenvolvidas pelos alunos.
Com a finalidade inversa de testar as aplicações AS desenvolvidas pelos alunos, existe em execução na máquina 'tejo' um servidor TCP concorrente (no porto 59000) que inicia localmente instâncias da aplicação *user* em resposta a acessos TCP por **netcat**, originados pelos alunos. Num acesso típico, os alunos especificam os parâmetros de execução remota da aplicação *user* na máquina 'tejo'.

Assim é possível executar em simultâneo na máquina 'tejo' instâncias da aplicação *user* que enviam mensagens a diferentes servidores AS alvo instalados na rede pública ou na rede do Técnico. Como por exemplo nas máquinas *sigma*.

Ambas as modalidades de teste acima referidas permitem aos alunos suportar tanto o desenvolvimento das suas aplicações como a componente de autoavaliação.

1.2 Instruções de activação dos testes

As aplicações *user* executadas remotamente na máquina 'tejo' vão ler as sequências de comandos a executar de *scripts* predefinidos, escolhidos pelos alunos para testar os seus servidores AS, mantendo separação de dados entre si não havendo assim qualquer interferência entre aplicações *user* em execução simultânea.

A figura que se segue ilustra uma sequência de teste típica.

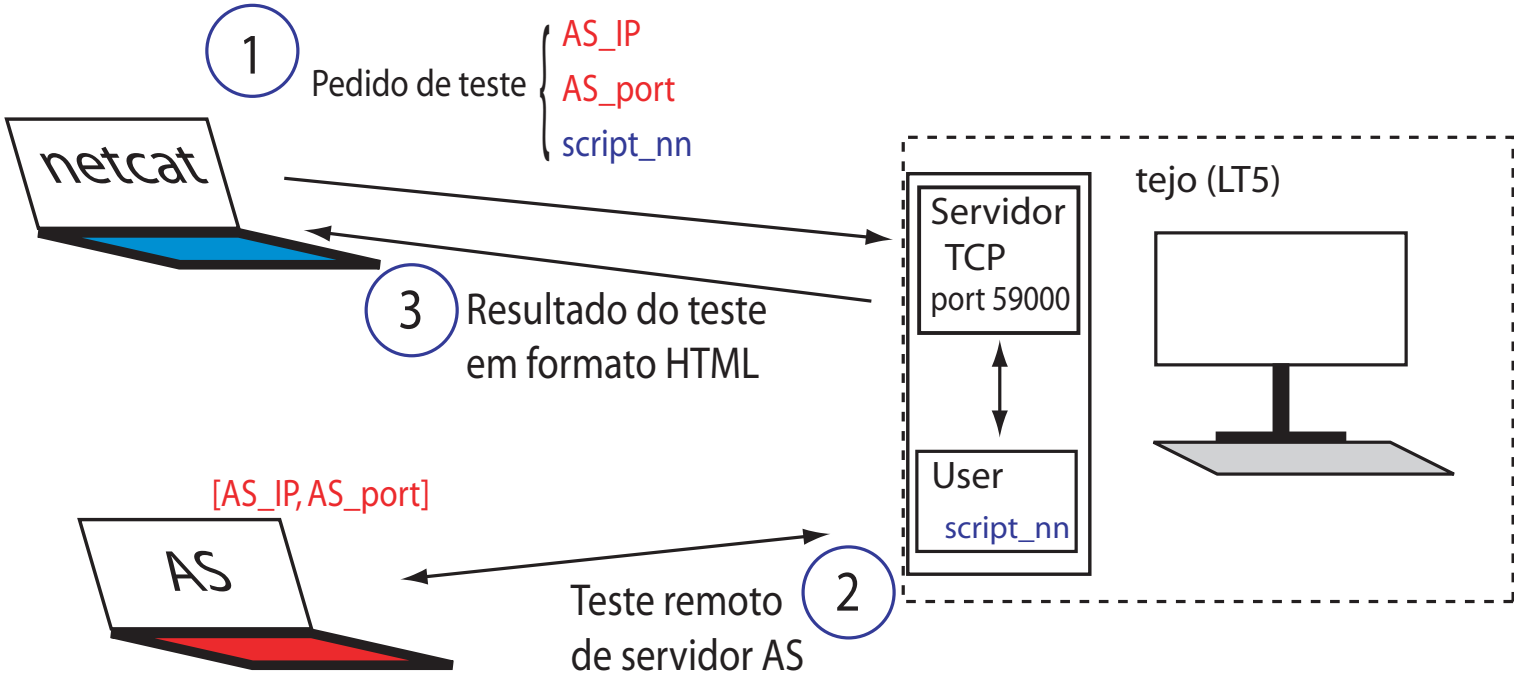

Figure 1: Teste remoto de servidores AS

O primeiro passo da sequência consiste em activar uma instância da aplicação *user* no 'tejo'. Nessa activação são especificados o endereço IP (AS_IP) e o porto (AS_port) do servidor AS a testar. São também especificados o número do *script* (script_nn) que contém os comandos a executar pela aplicação *user*, e o nome do ficheiro de relatório a ser guardado localmente no computador que activa o teste.
O segundo passo é a execução do *user* activado com os parâmetros acima referidos.

O terceiro passo consiste no envio pelo servidor de testes, para o computador usado na activação, do relatório de execução em formato html. Esse relatório pode ser visualizado usando um *browser* de internet.

Para executar um teste subordinado a um dado *script*, os alunos têm de compor e executar a seguinte linha de comando no terminal Linux:

echo "target_IP target_port script" | nc tejo.tecnico.ulisboa.pt 59000 > report.html

O texto entre "" constitui a **mensagem de comando** na qual, 'target_IP' é o endereço IP da máquina onde executam o seu servidor AS, 'target_port' é o porto da sua aplicação AS na referida máquina e 'script' é um inteiro com um ou dois dígitos que indica o número do *script* de teste que querem correr (ver *scripts* publicados na página da disciplina).
A linha acima descrita não deve ser segmentada. Isto é: Não deve ser executado primeiramente o comando **nc** e depois inserida a mensagem de comando já dentro do ambiente **nc**. Porque entre a aceitação da ligação TCP no servidor e a leitura da mensagem de comando existe uma reduzida tolerância de tempo para evitar o bloqueio de recursos no servidor.

Na mensagem com o formato apresentado acima, o servidor no tejo espera encontrar o caracter '\n' logo a seguir ao número do *script*, o qual é inserido pelo próprio comando 'echo' nas instalações *Linux* que serviram para elaborar este guia. Caso o comando 'echo' não insira o '\n' pode ser usado por exemplo o seguinte procedimento alternativo:

printf "target_IP target_port script\n" | nc tejo.tecnico.ulisboa.pt 59000 > report.html

Os espaços entre campos da mensagem de comando podem ser em qualquer número desde que o comprimento total da mensagem (incluindo '\n') não exceda os 25 bytes. Não serão aceites pedidos que definam como alvo os endereços IP público ou privado do tejo, ou começados por 127.

No final da execução, o servidor envia um ficheiro HTML com o resultado da referida execução obtido pela aplicação *user* cuja execução se solicitou. Esse ficheiro pode ser convertido em formato pdf para ser entregue no contexto da auto-avaliação do projecto.

O servidor AS alvo a testar pode ser executado no *sigma* cujo endereço IPv4 pode ser obtido executando o comando *hostname -i*. Note-se que o *sigma* pode ser endereçado por vários IPs. Caso queiram testar o servidor AS nos seus domicílios, os alunos devem proceder à configuração de *port forwarding* nos seus *routers* de acesso à rede.

Como exemplo, considere-se o servidor AS a testar no porto 58050 do *sigma* com IP=193.136.128.103 para execução do *script* número 6. A linha a executar numa máquina com Linux (podendo ser ela o próprio *sigma*) será:

echo "193.136.128.103 58050 6" | nc tejo.tecnico.ulisboa.pt 59000 > report.html

Ficando o relatório da execução guardado no ficheiro report.html na máquina na qual se executa o comando **nc**.

1.3 Descrição dos *scripts* de teste

Os *scripts* de comando da aplicação *user* no tejo invocada remotamente, estão em regra numerados de acordo com o grau de complexidade dos testes que produzem. No entanto, os alunos podem solicitar a execução dos *scripts* pela ordem que entenderem conveniente para testarem aspectos particulares do funcionamento dos seus servidores.

Aconselha-se o estudo pormenorizado dos *scripts* antes das respectivas execuções para que sejam percebidas de forma adequada as funcionalidades que são testadas pelos mesmos e assim se possam interpretar convenientemente os resultados dos relatórios.
Apresenta-se a seguir uma descrição sumária dos *scripts* pela ordem natural de numeração dos mesmos e que pode ser útil para testar funcionalidades dos servidores com complexidade crescente.

1.4 Teste de funcionalidades simples suportadas exclusivamente em UDP

Os *scripts* numerados de 1 a 4 testam funcionalidades básicas do servidor AS exclusivamente através de comunicações UDP. Estas funcionalidades estão relacionadas com o registo e identificação dos utilizadores na base de dados do AS.

O *script_01* deve ser executado com a base de dados do AS vazia, e solicita ao AS o primeiro registo do utilizador 111111, a sua saída de sessão logo seguida da sua entrada em sessão de novo.

O *script_02* deve ser executado a seguir ao *script_01*, pois testa a tentativa de entrada em sessão do utilizador 111111 com password errada e a sua entrada em sessão de novo com a password correcta.

O *script_03* deve ser executado com o utilizador 111111 já registado.

O *script_04* destina-se a testar o primeiro registo e a redefinir a password de utilizador para dois utilizadores. Deve ser executado com a base de dados no seu estado inicial, sem utilizadores, com os utilizadores 111111 e 222222 a efectuarem o seu primeiro registo. Uma boa execução deste *script* +e indicativa das funcionalidades básicas que o servidor deve possuir nas operações de **login**, **logout** e **unregister**.

1.5 Teste de funcionalidades básicas suportadas em TCP

Os testes que se seguem, (*scripts* 5 a 9) quando executados em sequência pela ordem proposta abrangem as capacidades básicas do servidor no que se refere à criação de leilões, recuperação de ficheiros de activo (*download*) e licitações. As operações envolvendo ficheiros de activo referem-se a ficheiros previamente disponibilizados na página da disciplina. Nas operações de licitação pressupõe-se que aquelas que forem admissíveis ocorrem dentro do tempo de vida do leilão.
Aconselha-se um teste que consiste na execução sequencial e rápida dos cinco scripts por forma a que a primeira execução do *script_09* ainda encontre os leilões abertos. Depois aguarda-se pelo tempo necessário a que os leilões em causa encerrem e executa-se de novo o *script_09*.

O *script_05* testa a abertura de um leilão numa base de dados sem leilões. Assim o leilão criado ficará com AID==001, para posteriores testes. O ficheiro referido como A.txt contém apenas 1 byte e encontra-se no pacote de ficheiros já disponibilizado na página da disciplina. Pressupõe o sucesso da abertura de sessão pelo utilizador 111111.

O *script_06* testa a recuperação pelo *user* (*download*) do ficheiro associado ao leilão 001 criado pelo *script_05*. A recuperação de ficheiros de activo pode ser efectuada a todo o tempo, independentemente de o leilão estar ou não activo.

O *script_07* testa a abertura de um leilão com um ficheiro de activo de grande dimensão. O leilão criado ficará com AID==002, para posteriores testes. O ficheiro referido como Whistlers_Mother.jpg contém 8.000.540 bytes e encontra-se no pacote de ficheiros já disponibilizado na página da disciplina. Pressupõe o sucesso da abertura de sessão pelo utilizador 222222.

O *script_08* testa a recuperação pelo *user* (*download*) do ficheiro associado ao leilão 002 criado pelo *script_07*. A recuperação de ficheiros de activo pode ser efectuada a todo o tempo, independentemente de o leilão estar ou não activo.

O *script_09* testa uma variedade de operações de licitação, com diferentes validades, em ambos os leilões constituídos pelos *scripts* 5 e 7. Pode ser executado quer durante o período de vigência de ambos os leilões, quer após o fim dos respectivos prazos.

1.6 Testes de consulta de leilões

Os *scripts* 10 e 11 são orientados para o teste de pedidos de listagens e de consultas com o comando **show_record**. Cada um destes testes deve ser iniciado com a base de dados completamente limpa.

1.7 Testes de temporização e encerramento de leilões

Os testes propostos nesta secção (*script_12*, a *script_15*) são orientados para a aferição das funcionalidades de temporização associadas ao prazo de validade dos leilões e ao fecho antecipado dos mesmos. Os *scripts* 12 e 13 mostram os efeitos da temporização através do pedido de listagens e de consultas com o comando **show_record**, respectivamente. Os *scripts* 14 e 15 envolvem a funcionalidade de encerramento intencional de leilões verificável através de listagens e de consultas com o comando **show_record**, respectivamente.
Cada um dos testes deve ser iniciado com a base de dados completamente limpa. Os testes com os *scripts* 12 e 13 são os mais demorados com uma duração típica de cerca de 30 segundos.

1.8 Testes de concorrência

Os testes propostos nesta secção estão orientados para avaliar as capacidades de concorrência do servidor. Eles baseiam-se em dois conjuntos de quatro *scripts* cada. O primeiro conjunto, constituído pelos *scripts* numerados de 21 a 24, idênticos, serve para testar a concorrência do servidor com quatro aplicações *user* em acesso simultâneo para abertura de 20 leilões cada um.
O *script_25* serve de base à constituição no servidor de uma base de dados para testar concorrência na descarga de ficheiros com o comando **show_asset**. A concorrência será testada pelo segundo conjunto de *scripts* numerados de 26 a 29.

Para testar a concorrência em abertura de leilões, deve garantir-se em primeiro lugar que a base de dados do AS está vazia, e que tem quatro utilizadores com os seguintes pares UID/password registados: 111111/aaaaaaaa, 222222/bbbbbbbb, 333333/cccccccc e 444444/dddddddd. De seguida abrem-se quatro terminais e neles executam-se os comandos:

echo “TargetIP TargetPort 21” | nc tejo.ist.utl.pt 59000 > report1.html

echo “TargetIP TargetPort 22” | nc tejo.ist.utl.pt 59000 > report2.html

echo “TargetIP TargetPort 23” | nc tejo.ist.utl.pt 59000 > report3.html

echo “TargetIP TargetPort 24” | nc tejo.ist.utl.pt 59000 > report4.html

As aplicações *user* no ‘tejo’ “percebem” pela numeração dos *scripts* envolvidos que devem arrancar em simultâneo. Portanto, só ao quarto pedido de execução é que todas arrancam. Recomenda-se contudo rapidez entre as primeira e quarta execuções por forma a não ocupar recursos por tempo excessivo.

Os resultados dos quatro testes encontram-se nos relatórios produzidos os quais devem ser analisados pois contêm a informação relevante para aferir das capacidades de concorrência do servidor. Estes quatro testes simultâneos podem servir para aferir a capacidade do servidor para resolver colisões de acesso à base de dados do AS em operações de escrita.

Para testar a concorrência do servidor AS em operações de descarga de ficheiros, deve a base de dados encontrar-se limpa e com o utilizador UID/password==111111/aaaaaaaa registado.

A execução do *script_25* serve para povoar a base de dados com os 19 leilões já conhecidos dos alunos e que constituem a base de dados fixa do AS que está instalado no ‘tejo’.

Após a execução deste *script*, podem ser executados em simultâneo os restantes quatro numerados de 26 a 29 tal como foram executados os *scripts* numerados de 21 a 24, *mutatis mutandis*. Os *scripts* numerados de 26 a 29 são parecidos entre si embora com diferentes ordenações. Após a descarga de cada ficheiro, o mesmo é comparado com o original para aferir do bom funcionamento das operações de *upload* e *download*.

Recomenda-se compartimentar os testes rigorosamente tal como exposto acima. O primeiro teste de concorrência deve referir sempre todos os números de *script* 21 a 24 e apenas estes. Do mesmo modo se deve proceder relativamente ao segundo teste de concorrência.

Proceder de outra forma conduz a testes que ocupam ou bloqueiam de forma improdutiva os recursos do ‘tejo’.