

Instituto Politécnico de Beja

Linguagens de Programação Dinâmicas

Enunciado de Trabalho Individual
2018-2019

Armando Jesus Ventura / José Caeiro

12 de Outubro de 2018

1 Aplicação de Segurança Informática

Os alunos têm o objetivo de realizarem uma aplicação de segurança informática construída com uma linguagem de programação dinâmica: Python; Ruby; Perl ou PHP. Recomenda-se o uso de Python. A escrita de código adicional na linguagem de programação C/C++ com o eventual uso de bibliotecas adicionais é possível e eventualmente desejável.

Os alunos devem realizar uma ferramenta de segurança informática que:

- permita detetar que portos de rede se encontram disponíveis numa ou mais máquinas remotas;
- permita determinar que conexões se encontrem ativas numa determinada máquina;
- processar ficheiros de log dum sistema Linux tentando obter informação sobre as tentativas indevidas de acesso de pelos menos 2 serviços (ex: http, SSH), nomeadamente:
 - listas de origem por país;
 - datas e horas de tentativas;
 - etc.
- outras características de segurança informática podem ser incorporadas no trabalho sendo valorizadas casuisticamente.

A ferramenta pode ser projetada para uso em modo de linha de comando ou modo gráfico dando-se preferência ao modo de linha de comando.

A aplicação deve permitir a produção de relatórios de segurança em formato PDF. Listas de informação devem ser produzidas em formato CSV. Deve poder ser produzida e processada informação para uma base de dados SQLite.

Alguns elementos de interesse para esta aplicação:

- uso das bibliotecas de rede de Python ou outras linguagens de programação dinâmicas;
- uso da biblioteca libpcap;
- a geração de PDFs com reportlab;

- gráficos com estatísticas, nomeadamente usando a biblioteca matplotlib;
- mapas geográficos usando as bases de dados GeoIP;
- autenticação no acesso à aplicação e aos *scripts*;
- cifragem dos dados, dados armazenados em ficheiros ou sqlite;

A aplicação projetada pode usar bibliotecas e ferramentas externas à aplicação.

2 Modo de Funcionamento

O trabalho, em todas as vertentes, deve ser efetuado em cima de um sistema de controlo de versões (ex: mercurial, svn, git, etc...). O tema do trabalho é idêntico para todos os alunos. A avaliação é individual.

2.1 Sistema Operativo

O sistema operativo escolhido para a realização do trabalho pode ser um dos seguintes:

- BSD numa das suas variantes;
- Linux numa das suas variantes;
- OSX numa das suas variantes.

Sugere-se que o trabalho seja desenvolvido numa máquina virtual ligeira e com um editor de texto de sistema em modo de consola, por exemplo: vi, emacs, nano, etc. A utilização de ambientes integrados de desenvolvimento, vulgarmente conhecidos por IDE, ou de editores que necessitem de suporte gráfico é fortemente desencorajada.

2.2 Sistema de Controlo de Versões

O trabalho deve ser realizado com controlo de versões¹. O sistema pode ser escolhido pelo próprio aluno, no entanto na realização do trabalho deverá este ser acompanhado de um relatório dos “commits” do próprio sistema de versões utilizado.

2.3 Data e Modo de Entrega

A data limite para a entrega do trabalho é até à meia-noite do último domingo da última semana de aulas da disciplina. Não há tolerância com a data e hora de entrega pelo que se aconselha o planeamento atempado da submissão.

A entrega deverá ser através da plataforma moodle.

Os alunos devem entregar o trabalho num ficheiro .zip com a seguinte estrutura.

Nome ficheiro zip: **nomeAluno_nº_LPD.zip**

Ficheiro **nomeAluno_nº_LPD.zip** deverá conter:

¹ Vidé http://en.wikipedia.org/wiki/Version_control

relatório.pdf : correspondente ao relatório do trabalho;

src : diretório onde reside o código fonte da aplicação;

apresentação.pdf: correspondente à apresentação do trabalho;

manual.pdf: correspondente ao manual de utilizador da aplicação;

srcdoc.pdf: correspondente à documentação do código.

É **importante** que os alunos que pretendam ser avaliados na disciplina enviem uma mensagem no fórum do moodle a informar dessa pretensão. Será questionado a todos os alunos, através de um post, quais efetivamente irão entregar o trabalho para avaliação. Este post será inserido no moodle até uma semana antes da data limite de entrega.

Componentes da Avaliação

O trabalho é composto pelas seguintes parcelas que serão avaliadas:

E - aplicação e scripts executáveis;

C - código da aplicação;

R - relatório do trabalho;

U - documentação para o utilizador;

D - documentação do código da aplicação;

A - apresentação.

É importante salientar duas questões fundamentais:

- a originalidade e inventividade das soluções;
- a honestidade na realização e na atribuição dos créditos intelectuais.

2.3.1 Aplicação Executável

A funcionalidade geral da aplicação, a obediência, e eventual ultrapassagem dos requisitos será levada em conta. Fatores de robustez e segurança na utilização da aplicação também serão importantes.

2.3.2 Código da Aplicação

No código da aplicação, entre outros aspetos, serão tomados em conta:

- a clareza do código;
- a estrutura do código;

- os conhecimentos de programação específicos para cada linguagem;
- a utilização correta do sistema de controlo de versões;
- a originalidade das soluções propostas;
- a referência adequada das fontes de inspiração do código.

2.3.3 Relatório do Trabalho

O relatório deve seguir estritamente as normas do IPBeja sobre produção de trabalhos académicos. Os alunos devem consultar no *site* do IPBeja estas normas.

É totalmente imperativo não ultrapassar as 20 páginas e é tomada em conta, entre outros elementos:

- a qualidade técnica e científica da escrita;
- a utilização correta do português e/ou do inglês;
- a estrutura do documento;
- a utilização adequada das referências bibliográficas.

O formato final deve ser em PDF.

2.3.4 Documentação para o Utilizador

A documentação para o utilizador deverá ser de leitura fácil e tecnicamente correta. O formato final deve ser em PDF.

2.3.5 Documentação do Código

Encoraja-se a utilização de ferramentas de extração automática da documentação do código diretamente a partir do código fonte.

Exemplos de sistemas deste género estão disponíveis em

- https://www.ruby-toolbox.com/categories/documentation_tools;
- <http://www.naturaldocs.org/>;
- http://en.wikipedia.org/wiki/Comparison_of_documentation_generators;
- <http://sphinx-doc.org/>;
- <http://docs.python.org/2/library/pydoc.html>;
- <http://wiki.python.org/moin/DocumentationTools>.

A conversão de HTML para PDF é facilitada por ferramentas como:

- <http://html2pdf.fr/en/default;>
- [http://www.cyberciti.biz/open-source/html-to-pdf-freeware-linux-osx-windows-software/;](http://www.cyberciti.biz/open-source/html-to-pdf-freeware-linux-osx-windows-software/)
- [http://code.google.com/p/wkhtmltopdf/.](http://code.google.com/p/wkhtmltopdf/)

O formato final deve ser em PDF.

2.3.6 Apresentação

O aluno deverá entregar uma apresentação do trabalho em formato PDF. Apenas este documento será tido em conta na avaliação.

2.3.7 Classificação Final

A classificação final, F , será calculada por:

$$F = 25\% \times E + 25\% \times C + 30\% \times R + 5\% \times U + 10\% \times D + 5\% \times A.$$

E - aplicação executável;

C - código da aplicação;

R - relatório do trabalho;

U - documentação para o utilizador;

D - documentação do código da aplicação;

A - apresentação.

A classificação final é arredondada para unidades entre 0 e 20 valores.

O tema do trabalho de recurso e época especial é o mesmo da época normal e os moldes de entrega são os mesmos.