Aplicação de Segurança Informática (Linguagens de Programação Dinâmica)

Escola Superior de Tecnologia e Gestão de Beja
Mestrado em Engenharia de Segurança Informática
Linguagens de Programação Dinâmica
Gonçalo Béjinha 13428

26 de junho de 2015
Beja

***Abstract*—No âmbito da Unidade Curricular de Linguagens de Programação Dinâmica do Mestrado em Engenharia de Segurança Informática foi elaborada uma aplicação de segurança informática. A aplicação apresenta várias funcionalidades de análise, nomeadamente, efetua o scan de IP e de ligações ativas de uma máquina numa rede local, a aplicação permite também processar ficheiros log da *firewall*, mostrando os resultados através de mapas e gráficos, por ultimo a aplicação dispõe de funções que permite exportar os resultados obtidos nos vários módulos em diversos formatos, tais como: pdf, csv e sql.**

Python foi a linguagem de programação utilizada para o desenvolvimento da aplicação, recorrendo às suas imensas e poderosas bibliotecas disponíveis interligaram-se vários *scripts*, de modo a permitir executar a aplicação módulo a módulo, ou executar tudo através da interface gráfica.

Palavras-chave: *segurança, scan ip, python, pentest, linguagens de programção, firewall log.*

I. Introdução

Segurança informática está cada vez mais na moda em todo o mundo e no mesmo sentido, o desenvolvimento de novas aplicações de segurança informática, estas que se apresentam-se como um elemento essencial da segurança informática, as aplicações são cada vez mais multifuncionais e mais poderosas.

O desenvolvimento de novas ferramentas está diretamente relacionado com as linguagens de programação dinâmicas, é sem dúvida essencial conhecer e em grande parte dominar várias linguagens de modo a poder liga-las entre si e criar um projeto mais robusto e eficaz. Como exemplo de linguagem de programação dinâmica demonstram-se neste projeto alguns *scripts* escritos em Python.

Este relatório irá revelar todas as bibliotecas utilizadas na criação da aplicação de segurança informática proposta na unidade curricular de Linguagens de Programação Dinâmicas, do Mestrado em Engenharia de Segurança Informática e ao mesmo tempo explicar o seu funcionamento.

II. Aplicação de segurança informática

A aplicação de segurança informática aqui apresentada foi desenvolvida com auxílio da linguagem de programação Python, no Sistema Operativo Debian 7.1 32bit, a aplicação tem como vários objetivos:

- Deteção dos IP numa rede local (IP Scan);
- Deteção de ligações ativas numa determinada máquina (Ligações ativas);
- Processamento de ficheiros log da *firewall*, (projeção das localizações das máquinas, apresentação de estatísticas) (Firewall data);
- Exportar informação em vários formatos, nomeadamente em pdf, csv, sql.

A aplicação apresenta uma interface gráfica com um sistema de autenticação, mas todos os módulos podem ser executados individualmente através da linha de comandos.

A. *Login/Interface*

Para abrir a aplicação em modo gráfico é preciso correr o ficheiro login.py, de seguida irá surgir uma janela para introduzir o *Username* e *Password*.

O script login.py, utiliza a biblioteca Tkinter [4], que permite criar aplicações gráficas.

Tkinter é uma biblioteca relativamente fácil de usar e apesar das suas limitações, permite criar interfaces com todos os elementos necessários para o correto funcionamento duma aplicação.

Figura 1 - Sistema de Autenticação

A interface gráfica principal foi também desenvolvida com recurso a Tkinter, e para exemplificar a utilização desta biblioteca apresenta-se o seguinte código que irá criar uma janela com botão informativo.

Figura 2 - Interface gráfica principal

Exemplo 1 – Tkinter janela com caixa de texto

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Developer: Goncalo Bejinha
import Tkinter
import tkMessageBox

top = Tkinter.Tk()
top.title("LPD 2015")
def LPDInfo():
      tkMessageBox.showinfo("LPD Aplication", "hello world")
B_info = Tkinter.Button(top, text = "SOBRE A APLICAÇÃO",
command = LPDInfo)
Area = Tkinter.Canvas(top, height=150, width=150)
B_info.pack()
Area.pack()
top.mainloop()
```

Depois de efetuar o login corretamente, será apresentada a janela principal, onde se destaca os botões Figura 2:

- **Sobre a aplicação**

Quando clicado este botão será apresentado um pequeno texto introdutório sobre a Aplicação.

"No âmbito da disciplina de Linguagens de Programação Dinâmica do Mestrado em Engenharia de Segurança Informática foi desenvolvida a presente aplicação, que permite detetar os portos ativos nas máquinas de uma rede local, analisar ficheiros log e salvar relatórios dos dados."

- **1 – IP Scan**

O Botão "1 – IP Scan" quando clicado irá executar o módulo lan_scan.py, que permite fazer o scan a uma rede local e detetar o IP das máquinas.

- **2 – Portos Ativos**

O Botão "2 – Portos Ativos" quando clicado irá executar o módulo active.py, que permite fazer o scan das ligações ativas de uma determinada máquina.

- **3 – Firewall**

O Botão "3 – Firewall" quando clicado irá executar o módulo FirewallData.py, que permite analisar os ficheiros log, apresentando no mapa-mundo os locais dos ataques detetados na *firewall*, este botão apresenta também um gráfico de barras com o número de ataques de cada país.

B. IP Scan

O botão "1 – IP Scan", corresponde ao módulo lan_scan.py do projeto.

Para executar corretamente este código o utilizador precisa de indicar a gama de IP´S que quer analisar, por exemplo: python lan_scan.py –network=192.168.1.0/24

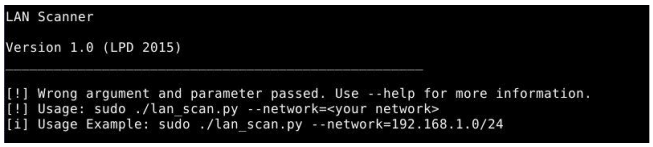

Figura 3 - Ip Scan em execução

Este módulo realiza o scan das máquinas ligadas à rede local, mostra o MAC e o IP das respetivas máquinas e cria um pequeno relatório dos resultados obtidos.

Para o correto funcionamento deste código foi preciso instalar Scapy[6].

Scapy permite analisar os pacotes que circulam na rede e desse modo mostrar os *MAC address* das máquinas e os seus *IP´S*.

Exemplo 2 – import lan_scan.py

```
import sys,  getopt
from scapy.all import srp,Ether,ARP,conf
from time import gmtime, strftime
```

C. Ligações ativas

active.py é um dos módulos mais importantes do projeto, pois irá analisar as ligações ativas de uma determinada máquina, este código apresenta funcionalidades idênticas à ferramenta NMAP[9][12], analisando os portos TCP e UDP Figura 4.

Para executar corretamente este código o utilizador precisa de indicar o IP da máquina vitima que quer analisar e os argumentos opcionais, por exemplo: python active.py -v -sS -sU -p 21,22,80,135-139,443,445 -t 192.168.1.128

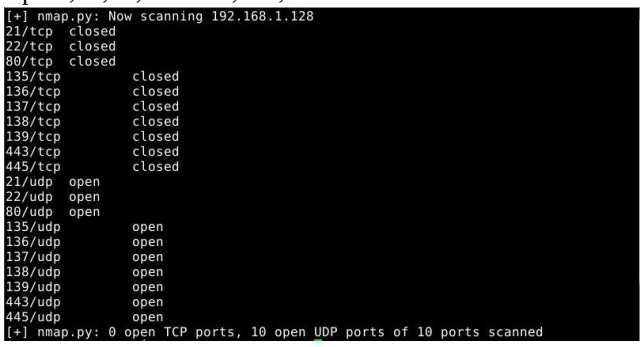

Figura 4 – Scan Portos Ativos

Exemplo 3 – Import active.py

```
# Copyright:    (c) phillipsme 2014

import socket
import argparse
import sys
```

D. *Firewall data*

A análise de ficheiros log é apoiada com a representação gráfica, através do módulo FirewallData.py, este ficheiro também não foge à exceção, ou seja, pode ser executado pela linha de comandos ou pela interface gráfica, clicando no botão "3 – Firewall".

A informação dos ficheiros log é tratada de modo a ser recolhida a informação essencial para se saber a localização dos ataques e o respetivo número de ataques que cada país apresenta.

Para executar o script deve entrar na pasta src e escrever o seguinte comando: python FirewallData.py

Para o correto funcionamento deste módulo foi utlizada a biblioteca de maplotlib[13][16][17], que permite a projeção de mapas e neste caso apresentar a informação tratada no respetivo mapa.

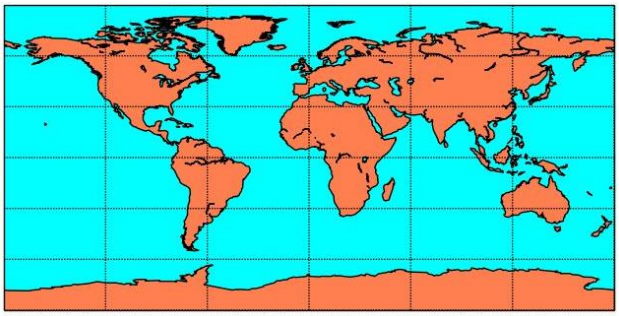

Figura 5 - Projeção da localização ficheiro log

E. *Exportar informação*

A aplicação permite gravar em vários formatos, para visualizar os resultados obtidos dos módulos.
No final de cada módulo foi adicionada uma porção de código, de modo a facultar ao utilizador as opções de exportar os resultados em formato PDF, CSV ou SQL.

III. Relatórios da aplicação

Como documentação de apoio e análise do código utilizado no projeto foi elaborado um relatório que juntou toda a informação de todos os módulos, o relatório foi criado com pydoc, programa que gera documentação em formato de texto ou html.

Pydoc[2] é muito útil e fácil de utilizar, por para criar um documento sobre o módulo login.py o utilizador deve usar o seguinte comando: pydoc login

Figura 6 - pydoc Sourcedoc

IV. Sistema de controlo de versões

De modo a apoiar o desenvolvimento da aplicação de segurança informática, utilizou-se um sistema de controlo de versões sendo inicialmente utilizado o serviço do Google Code, mas na data da conclusão deste projeto foi preciso exportar todos os documentos para o serviço GitHub de forma a ser possível executar uma verificação de versões do projeto.
Para completar este relatório serão aqui apresentados alguns comandos importantes para utilização do git.

- git init – iniciar um novo repositório
- git add nomedoficheiro.extensão – adicionar ficheiros ao stage (local temporário que armazena a referência para os ficheiros)
- git status
- git commit –m "mensagem sobre alterações"

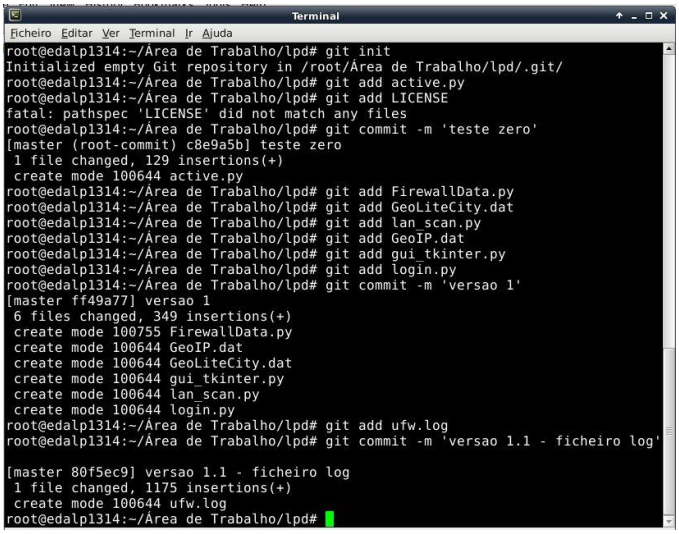

Figura 7 - git em execução

V. Aspetos Positivos e negativos

Durante o desenvolvimento da aplicação ficou comprovada a facilidade de obtenção de código de outros autores, através de repositórios como o Google code ou GitHub, por esta razão a aplicação tornou-se numa ferramenta automatizada que iria interligar várias outras pequenas ferramentas.

A. *Aspetos positivos*

- A interface gráfica é fácil de utilizar e explorar, pois não requer qualquer conhecimento técnico
- Portabilidade da aplicação para as diferentes plataformas (Linux, Mac, Windows)
- Os módulos podem ser executados individualmente, desde que estejam as bibliotecas necessárias instaladas
- Aplicação apresenta diversas e importantes funções na área da segurança informática (Scan de IP, Ligações Ativas, Processamento de log)

B. *Aspetos negativos*

- Utilização de poucas linguagens de programação
- O código da aplicação pode conter lixo ou algumas falhas

- Falta de alguns módulos para aumentar a complexidade e funcionalidades da aplicação
- Código pertencente a outros autores.

VI. CONCLUSÃO

Terminada a fase de desenvolvimento da aplicação de segurança informática, o autor deste projeto sublinha que foi um trabalho muito enriquecedor, quer ao nível da escrita de código em várias linguagens de programação, matérias que ajudaram e obrigaram o aluno a praticar e aprofundar os seus conhecimentos nas linguagens utilizadas para o desenvolvimento da aplicação, foi também enriquecedor ao nível do conhecimento técnico teórico da segurança informática no que diz respeito a questões de bibliotecas e interligar vários scripts das mais variadas linguagens de programação dinâmicas.

Um outro aspeto positivo a destacar durante a elaboração deste projeto, foi a facilidade de obter informação útil e mais precisamente, porções de código disponível na internet que ajudaram a melhorar cada vez mais os módulos individuais.

Por outro lado, um dos aspetos negativos a salientar neste projeto, prende-se com a pouca prática em programação e mais precisamente em aplicações de segurança informática.

A aplicação no futuro pode ser facilmente melhorada, quer no código existente nos módulos existentes, quer em outros módulos que possam vir a ser adicionados com outras funcionalidades.

REFERÊNCIAS

[1] webnull, "scan-network," 14 Agosto 2011. [Online]. Available: https://github.com/webnull/scan-network. [Acedido em 22 junho 2015].

[2] Vivek, "Html to Pdf," 3 Outubro 2012. [Online]. Available: http://www.cyberciti.biz/open-source/html-to-pdf-freeware-linux-osx-windows-software/.

[3] R. rosario, "Quick and Dirty GeoIP Lookup Function in Python," [Online]. Available: http://rickyrosario.com/blog/quick-and-dirty-geoip-lookup-function-in-python/. [Acedido em 23 junho 2015].

[4] Python, "Tkinter - Python interface to Tcl/Tk," Python Software Foundation, [Online]. Available: https://docs.python.org/2/library/tkinter.html.

[5] Python, "Pydoc," [Online]. Available: http://pydoc.org/2.4.1/pydoc.html. [Acedido em 25 Junho 2015].

[6] Python, "Modules," Python Software Foundation, [Online]. Available: https://docs.python.org/2/tutorial/modules.html. [Acedido em 23 junho 2015].

[7] Python, "Cryptographic modules for Python," Python Software Foundation, 20 junho 2014. [Online]. Available: https://pypi.python.org/pypi/pycrypto.

[8] Maxmind´s, "Pure Python GeoIP API," [Online]. Available: https://pypi.python.org/pypi/pygeoip/.

[9] Maxmind, "GeoLite Legacy Downloadable Databases," MaxMind Dev, [Online]. Available: http://dev.maxmind.com/geoip/legacy/geolite/.

[10] B. Matplotlib, "Equidistant Cylindrical Projection," [Online]. Available: ttp://matplotlib.org/basemap/users/cyl.html. [Acedido em 21 junho 2015].

[11] J. Hunter, "matplotlib," [Online]. Available: http://matplotlib.org/.

[12] gambit240809, "PortScanner," 19 novembro 2013. [Online]. Available: https://github.com/OpenFireTechnologies/PortScanner. [Acedido em 22 junho 2015].

[13] P. Biondi, "Scapy - Download and Installation," Python, [Online]. Available: http://www.secdev.org/projects/scapy/doc/installation.html.

[14] "The Reportlab Toolkit," Python Software Foundation , [Online]. Available: https://pypi.python.org/pypi/reportlab. [Acedido em 23 junho 2015].

[15] "Python-nmap : nmap from python," [Online]. Available: http://xael.org/norman/python/python-nmap/.

[16] "Python for PDF Generation," Devshed Network, [Online]. Available: http://www.devshed.com/c/a/Python/Python-for-PDF-Generation/#whoCFCPh3TAks368.99. [Acedido em 25 junho 2015].

[17] "Numerical Python," [Online]. Available: http://sourceforge.net/projects/numpy/files/OldFiles/1.4.0rc2/.

[18] "How to detect the country and city of a user accessing your site?," stackoverflow, [Online]. Available: http://stackoverflow.com/questions/1163136/how-to-detect-the-country-and-city-of-a-user-accessing-your-site. [Acedido em 24 junho 2015].