ACADEMIA
NACIONAL DE CIBERSEGURANÇA

Trabalho Final Anciber

Programa de Scan de Redes – The Scanner

Elaborado por:

Samuel Viegas

Aluno 2ª Edição

Índice

Introdução

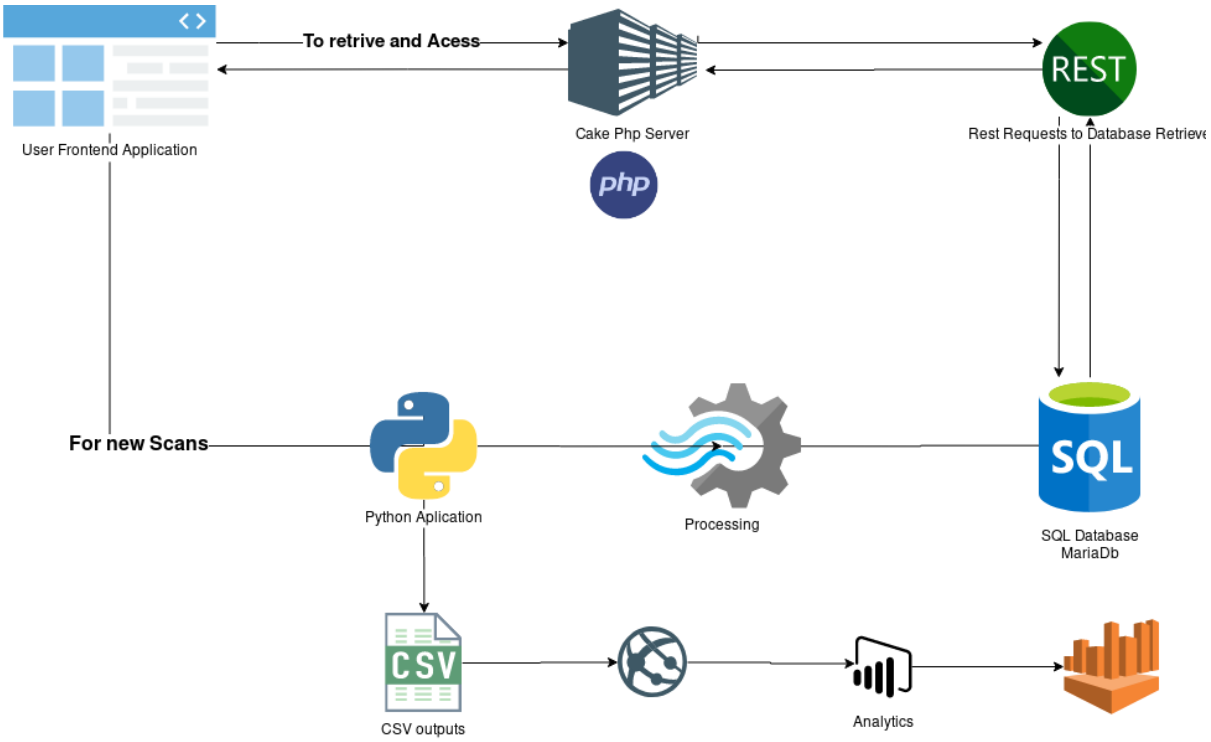
O objetivo do trabalho era construir um programa de scan de redes e ip's, com a funcionalidade de guardar os scans em base de dados e mostrá-los ao utilizador quando ele pedisse.

Com isso em mente, pensei numa solução que tem as seguintes catacteristicas:
Faz uso de python para backend. (e front-end com tkinter!)
Faz uso de sql para base de dados.
Faz uso de bash scripting para instalar o projeto.
Faz uso de php e html - cake php - para front end e mostrar os resultados ao utilizador.

Além disso, o programa exporta ficheiros .csv que podem ser posteriormente usados em programas de visualização de dados.

O programa tem como função e objetivo descobrir todos os ips, Sistemas operativos, kernels, versão dos kernels, serviços, versão de serviços de um ip ou uma rede.

O workflow de todo o trabalho foi o seguinte:



Características e elementos - geral

O trabalho é constituído por 4 grandes elementos:

1. Ferramenta em python que realiza os scans e reporta os seus resultados no terminal, em ficheiros e guarda na base de dados;

2. Base de dados em sql que guarda os scans realizados;

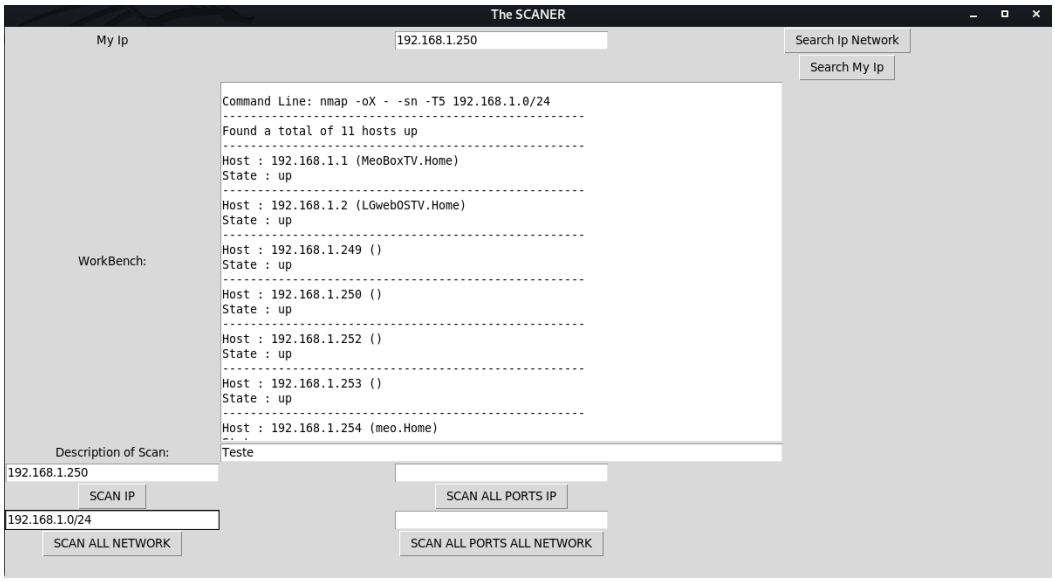
3. Framework CakePHP que mostra os resultados guardados na base de dados ao usuário;

4. Bash scripts que realizam as configurações, instalam as dependências e facilitam ao máximo a vida do utilizador ao usar o programa;

Análise ao elementos

1º elemento – Ferramenta Python “The Scanner”:

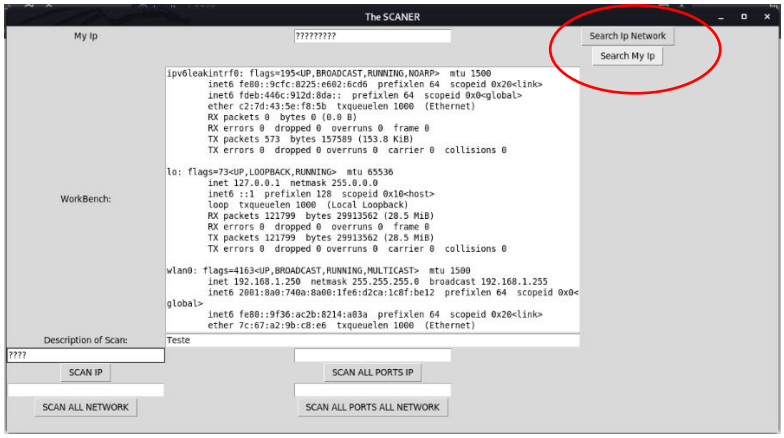
A ferramenta - The Scanner - é constituída por 4 scans, 1 workbench e 2 botões de ajuda.



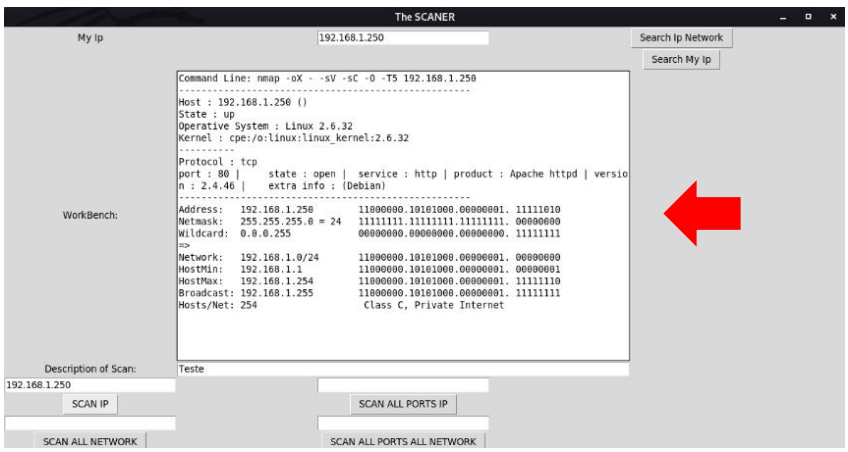
Os scans são:
1- Scan rápido de um IP: faz scan das portas mais normais de um ip de forma rápida.
2- Scan de todas as portas de um IP: faz scan de todas as portas de um ip.
3- Scan rápido de rede: faz scan de todos os ips de uma rede e reporta os que estão up.
4- Scan das portas de todos os equipamentos numa rede.
Nota: Este é o único que não reporta um ficheiro por causa do uso de threads para a maior velocidade e eficiência. É reportado o resultado no terminal.

Quase todos são guardados na base de dados, exceto o 4º modo de scan, devido á dificuldade de escrita de threads num ficheiro.
O último, que faz scan de toda a rede e das portas desses equipamentos foi feito com threads para melhorar o desempenho.

Os botões de ajuda:
os botões de ajuda permitem saber o ip e saber o endereço de rede de um qualquer ip.



O workbench:
No workbench é sempre acrescentado cada scan ou ação de ajuda que se faz.
Nota: Para manter a higiene do workspace é necessário apagar regularmente algum conteúdo da janela.



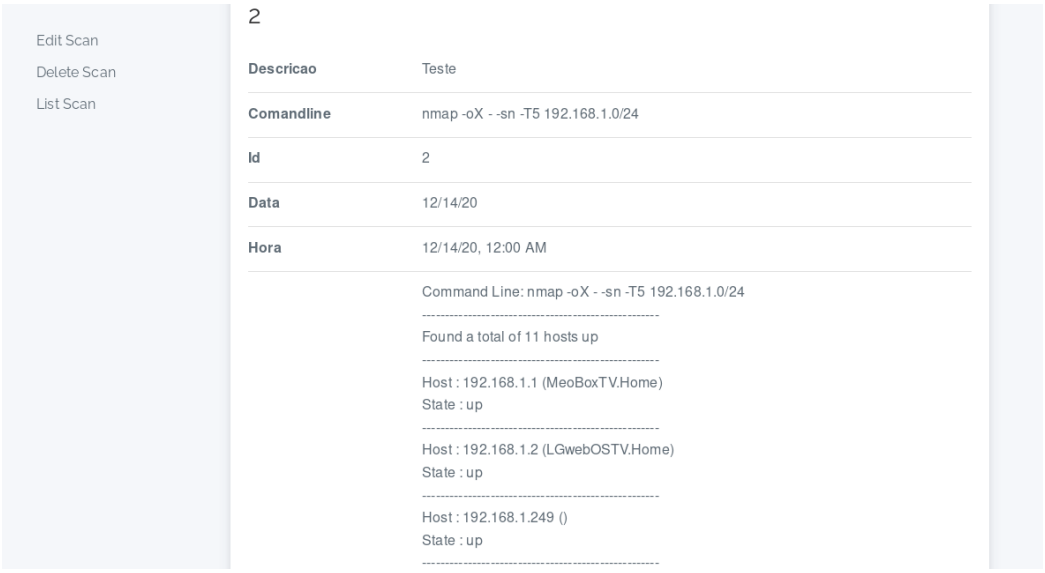
2º elemento – Base de dados SQL:

A base de dados é criada em sql , mais precisamente usando um servidor mariadb.
É instalado o adminer para facilitar o uso da base de dados.

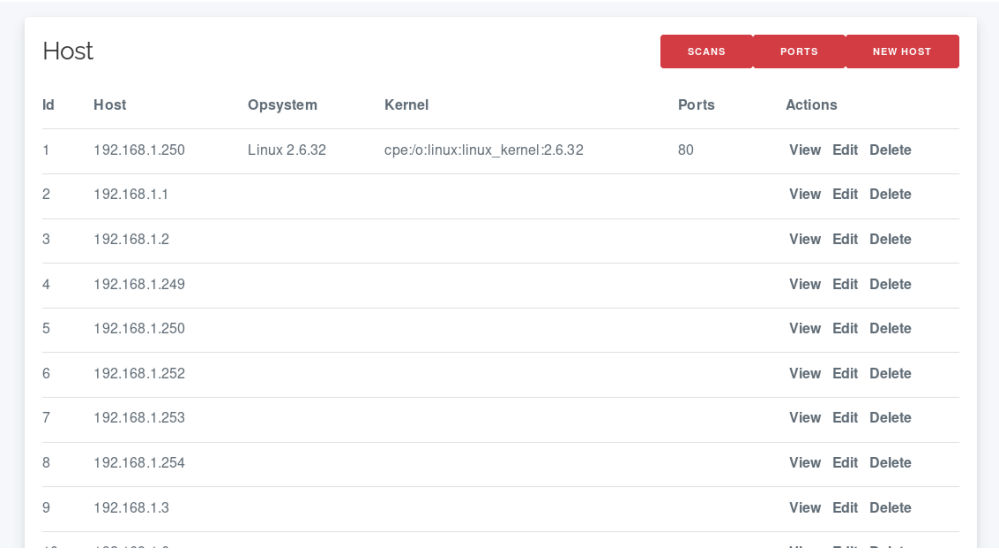
A base de dados tem 3 tabelas :
scan - são os scans feitos
host - são os hosts descobertos
port - são os portos descobertos

3º elemento – Cake PHP:

É usado o Cake PHP como front-end para mostrar os resultados ao utilizador.
É possível ver todos os scans efetuados.
É possível ver o resultado de um scan feito ao clicar em 'View scan'. Mostra o conteúdo do ficheiro de report guardado.



Também é possível ver todos os host e ports e cada respetiva informação clicando no respetivo 'View host/port'.



É possível editar qualquer elemento.
É possível eliminar qualquer elemento da base de dados : scans,hosts ou ports.
Pode-se criar hosts e port individuais, mas recomenda-se que seja a ferramenta a criá-los e adicioná-los.
Para criar scans é usada a ferramenta - The Scaner - que é logo lançada ao correr o script inicial.
Para lançar o The Scaner basta clicar em 'New Scan', no canto superior direito.

4º elemento – Bash Script:

Foi criado um script que instala as dependências necessárias e reduz ao máximo a interação e configuração que o utilizador tem que fazer.

Características e Elementos – em detalhe

Em seguida apresentam-se os elementos em maior detalhe de implementação.

1- Ferramenta Python “The Scanner”:

localizado em (/var/www/cakephp/app/webroot)

1- scanner.py

Nesta classe é implementada toda a parte gráfica e o ciclo de vida do programa. É a classe de topo. Importa e faz uso dos métodos criados nas outras classes. É onde são desencadeados os métodos ao clicar nos botões e onde são feitas as verificações. Para a parte gráfica é usada a biblioteca gráfica Tkinter.

2- ScanDefinitions.py

Nesta classe é feito tudo o que tem a ver com os scans propriamente ditos. É usado o módulo python-nmap que realiza scans por uso de um objeto python.
Foi usado um scanner normal e um continuo para melhorar a velocidade dos scans.
Os resultados são processados e são guardados os resultados mais interessantes.

3- sqlConnector.py

Nesta classe faz-se todas as conexões com a base de dados e todos as querys SQL.
É usado um mysql.connector e liga-se á base de dados do programa.
As querys estão preparadas e estão devidamente protegidas de SQL injections.

Esta hierarquia permite organizar melhor o trabalho e perceber melhor todas as funções que o programa tem e faz.

A ferramenta ainda pode melhorar esteticamente (foi a primeira vez a usar Tkinter).

2- Base de dados SQL

Como já foi dito, a base de dados é constituída por 3 tabelas: Scan, Host e Port
Cada scan tem as seguintes características: id,data,hora,descricao, comando executado e o ficheiro onde foi guardado o scan.
Cada host tem as seguintes características: id,host,sistema operativo, kernel e os portos abertos
Cada porto tem as seguintes características: id,numero,estado (aberto ou filtrado), serviço, versão do serviço e informações adicionais do serviço.
Os ids são incrementais por 1 e são chave primária.

O objetivo era usar chaves estrangeiras e conseguir ligar cada host e port ao respetivo scan, mas não recebi qualquer ajuda ao meter essa dúvida aos instrutores.

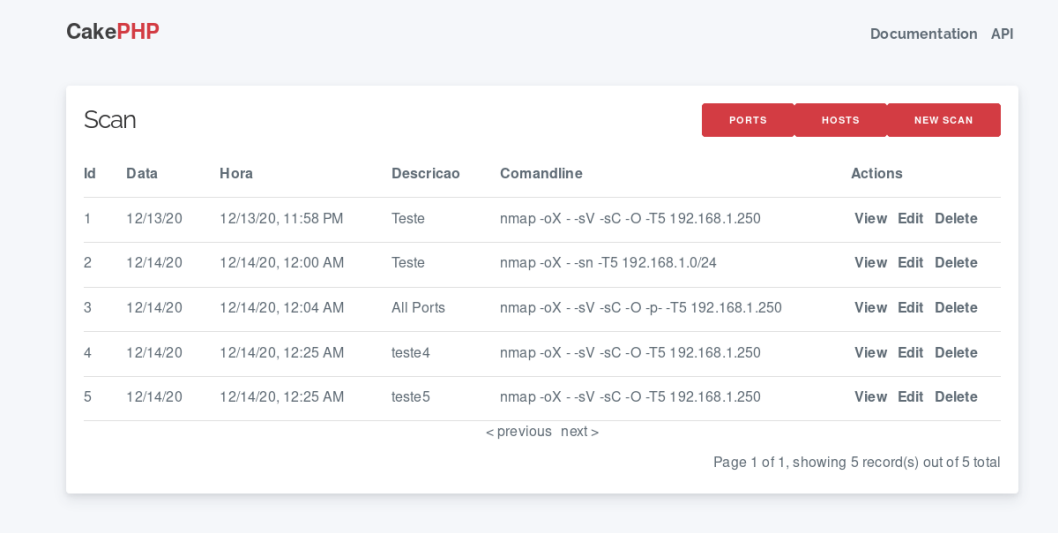
3- Resultados Cake PHP

Os Resultados são apresentados ao utilizador num web browser usando pedidos rest pela framework cake php.

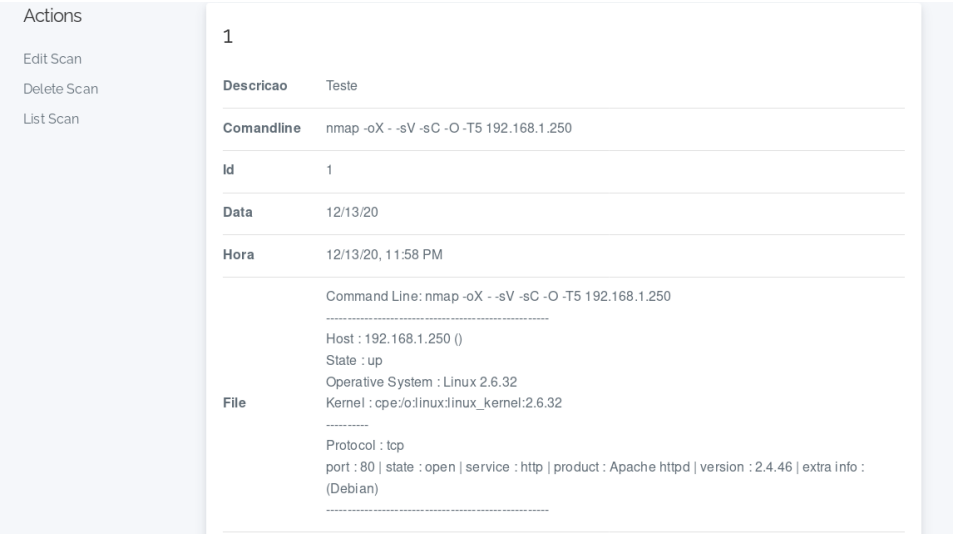
Após o deploy do server, o Cake redireciona o utilizador logo para a página de scans, após ele clicar no link mostrado no terminal.

Isto foi conseguido ao mudar na pasta config o ficheiro routes.php, metendo como a rota principal o controlador Scans para a view index.

É possível ver os Scans feitos:

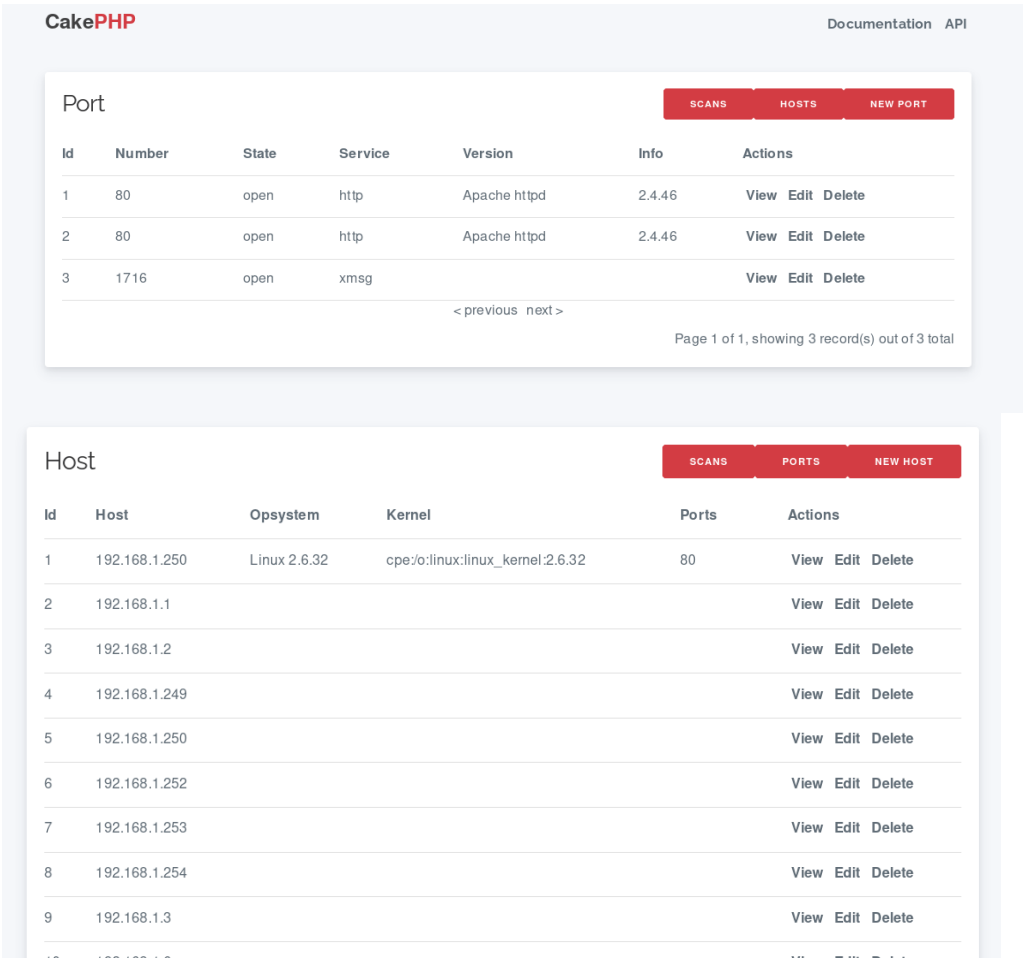


e ver cada scan individualmente:



Isto foi conseguido ao usar funções de ficheiros e ao ler o conteúdo do ficheiro do scan e mostrá-lo ao utilizador.

Perrmite ver os hosts e as portas descobertas:



Cada página dá para mudar para as outras facilmente, apenas clicando nos botões no canto superior direito.

Isto foi conseguido ao adicionar ao template um link para o controlador pretendido e para a ação index.

É permitido criar host e portas individuais, mas para os scans tem que se usar a ferramenta python.

A ferramenta para scans é iniciada assim que se corre o ficheiro de instalação.
Para iniciar a ferramenta outra vez basta carregar no botão "New Scan" ou iniciá-la noutro terminal.

Isto foi conseguido ao criar uma função no Controlador Scans que inicia o programa por meio de uma chamada ao sistema.

4- Bash Script – fast deploy: (bash em root)

Foi criado um script que instala as dependências necessárias.

Além disso, implementa ifs e ciclo while em bash.

Em detalhe, realiza as seguintes operações:

Instala automaticamente o php, o composer, o apache, o sql (mariadb), configura usando o mysql_secure_instalation, configura a base de dados usando o ficheiro em anexo nº2, instala o adminer, instala o python e dependências necessárias usando o ficheiro em anexo nº3 e desempacota e prepara o servidor e o programa.

É necessário correr em root devido ás instalações e configurações que tem de fazer.

Conclusão

Penso que consegui realizar um bom trabalho.

O trabalho final demonstra conhecimentos nos seguintes temas abordados durante a formação:

Bash scripting e linguagem bash – Script de instalação

Python – Ferramenta python

SQL e bases de dados – Manipulação de base de dados sql (mariadb), adminer e interação python-bd e interação bd-php

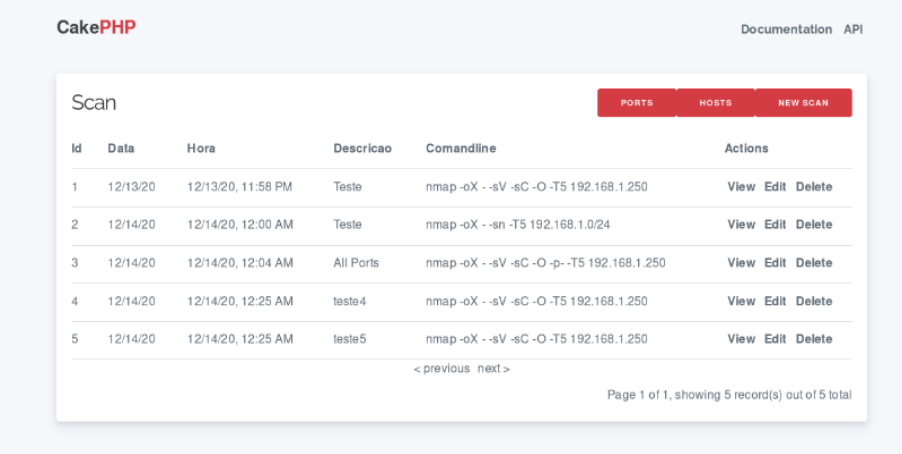
PHP e REST – Framework Cake PHP

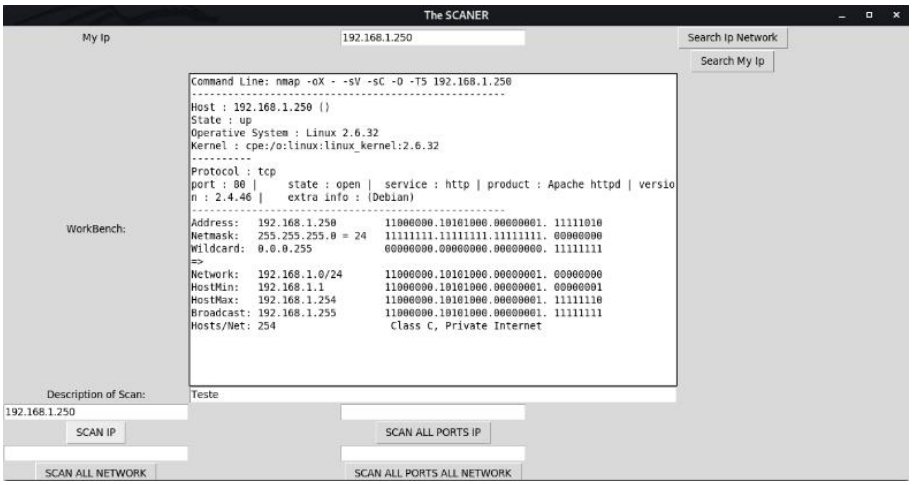
Scan de redes – nmap e modulo python python-nmap

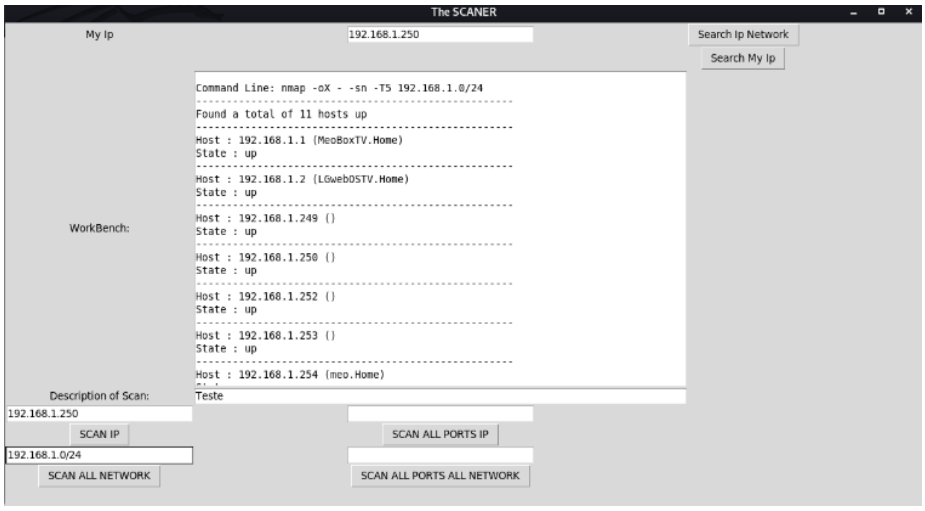
Cibersegurança e pentesting – Reconhecimento:escaneamento de redes, portas, serviços, versões

Anexo

1- Screenshots







2- Ficheiro de criação da base de dados

```
3- USE cakephp;
4- CREATE TABLE `host` (
5-   `id` int(11) NOT NULL AUTO_INCREMENT,
6-   `host` varchar(100) NOT NULL,
7-   `opsystem` varchar(100) DEFAULT 'Unknow',
8-   `kernel` varchar(100) DEFAULT 'Unknow',
9-   `ports` varchar(500) DEFAULT 'Unknow',
10-    PRIMARY KEY (`id`)
11- ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
12-
13-
14- CREATE TABLE `port` (
15-   `id` int(11) NOT NULL AUTO_INCREMENT,
16-   `number` varchar(10) NOT NULL,
17-   `state` varchar(100) DEFAULT 'Unknow',
18-   `service` varchar(100) DEFAULT 'Unknow',
19-   `version` varchar(100) DEFAULT 'Unknow',
20-   `info` varchar(100) DEFAULT 'Unknow',
21-   PRIMARY KEY (`id`)
22- ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
23-
24-
25- CREATE TABLE `scan` (
26-   `id` int(11) NOT NULL AUTO_INCREMENT,
27-   `Data` date DEFAULT NULL,
28-   `Hora` timestamp NULL DEFAULT current_timestamp() ON UPDATE
   current_timestamp(),
29-   `Descricao` varchar(300) DEFAULT 'No Description',
30-   `comandline` varchar(300) DEFAULT 'Unknow',
31-   `file` mediumblob NOT NULL,
32-   PRIMARY KEY (`id`)
33- ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

3- Ficheiro de instalação python

```
4- #!/bin/bash
5- #Python
6- apt-get install python
7- apt-get install pip
8- pip install python-nmap
9- pip install mysql-connector-python
10- apt-get install ipcalc
11- apt-get install nmap
```