

Trabalho Final Anciber

Programa de Scan de Redes – The Scanner



Elaborado por:

Samuel Viegas

Aluno 2ª Edição

Conteúdo

Introdução	3
Requisitos Mínimos.....	5
Características e elementos - geral.....	6
O trabalho é constituído por 4 grandes elementos:	6
1º elemento – Ferramenta Python “The Scanner”.....	6
2º elemento – Base de dados SQL:	8
3º elemento – Cake PHP:	8
4º elemento – Bash Script:.....	9
Características e Elementos – em detalhe.....	10
1- Ferramenta Python “The Scanner”:	10
1- scanner.py.....	10
2- ScanDefinitions.py	10
3- sqlConnector.py.....	10
2- Base de dados SQL.....	10
3- Resultados Cake PHP.....	11
4- Bash Script – fast deploy: (bash em root).....	13
Versão 2 (Correções ao código e ao relatório)	14
Corrigido Introdução, Conclusão e estrutura do relatório	14
Adicionado secção Requisitos Mínimos.....	14
Corrigido ACE (Arbitrary code execution)	14
Corrigido SQL Injection	14
Corrigido instalação de Apache desnecessária	15
Conclusão	16
Anexo.....	17
Screenshots.....	17
Ficheiro de criação da base de dados	18
Ficheiro de instalação python	18
FastInstallation	19

Introdução

O objetivo do trabalho é desenvolver toda uma infraestrutura para a compreensão e manipulação das mais usadas e cobiçadas estruturas da atualidade: as redes.

Desde as décadas de 1970 onde foi criada a Internet, no auge da guerra fria, no momento denominada ARPANET que este tipo de estruturas domina a tecnologia.

A ARPANET, mais tarde denominada INTERNET foi criada com o objetivo da troca rápida de informações secretas militares. Mais tarde foi aberta ao meio acadêmico e só a partir daí cresceu exponencialmente, com contributos das mais variadas personalidades e culturas. Foram formados consórcios e instituições reguladores para chegarem a acordos e a padrões standardizados e iguais para todos. Finalmente chegou ao que vemos hoje, a estrutura mais usada pela população e a base da sociedade atual e futura.

Atualmente pode observar-se a importância deste tema das redes com as redes neuronais, as redes sociais e o uso da Internet em geral – que é nada menos nada mais que um conjunto de redes interligadas entre si.

Com tudo isto em mente, surgiu a ideia de construir um programa de scan de redes e ip's de redes de internet, com a funcionalidade de guardar os scans em base de dados e mostrá-los ao utilizador.

Foi então elaborada uma solução que tem as seguintes características:

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 em web e/ou usados como ferramenta para análise de dados (esta última parte é totalmente opcional, o programa só fornece o output em csv).

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 toda a implementação é o seguinte:

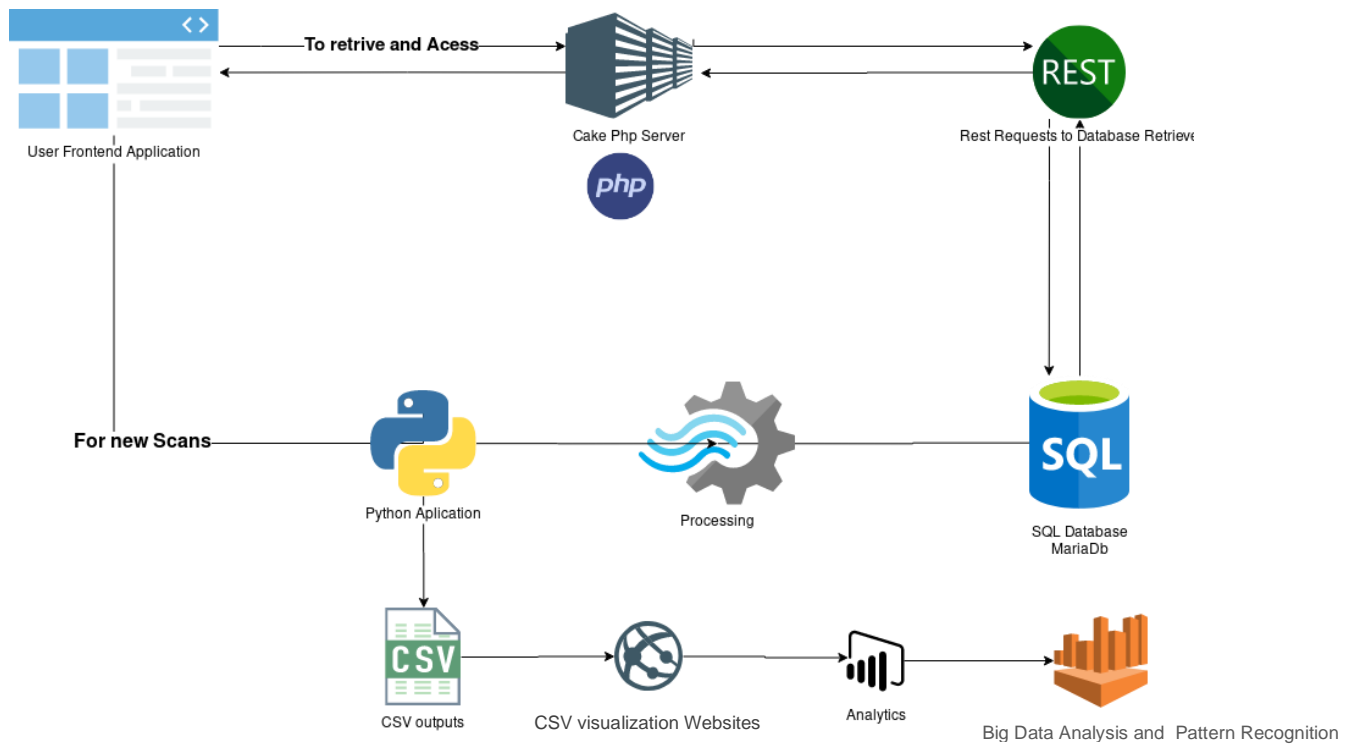


Figura 1-Workflow do Projecto Final

O programa apresenta uma GUI (guest user interface) e é por isso mais fácil de utilizar, não precisa de ser um programador experiente.

No entanto, só é possível usar o programa em sistemas baseados em Linux e por isso é preciso algumas bases tecnológicas ao usar o programa, umas das quais estar familiarizado com ambiente Linux.

O programa foi desenvolvido no Kali Linux versão:

Linux kali 5.7.0-kali1-amd64 #1 SMP Debian 5.7.6-1kali2 (2020-07-01) x86_64 GNU/Linux

Tentei abordar os seguintes temas:

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

Requisitos Mínimos

- Sistema Operativo Linux

Link para Download do Github:

https://github.com/SamTheSapien/Anciber_Trabalho_Final

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.

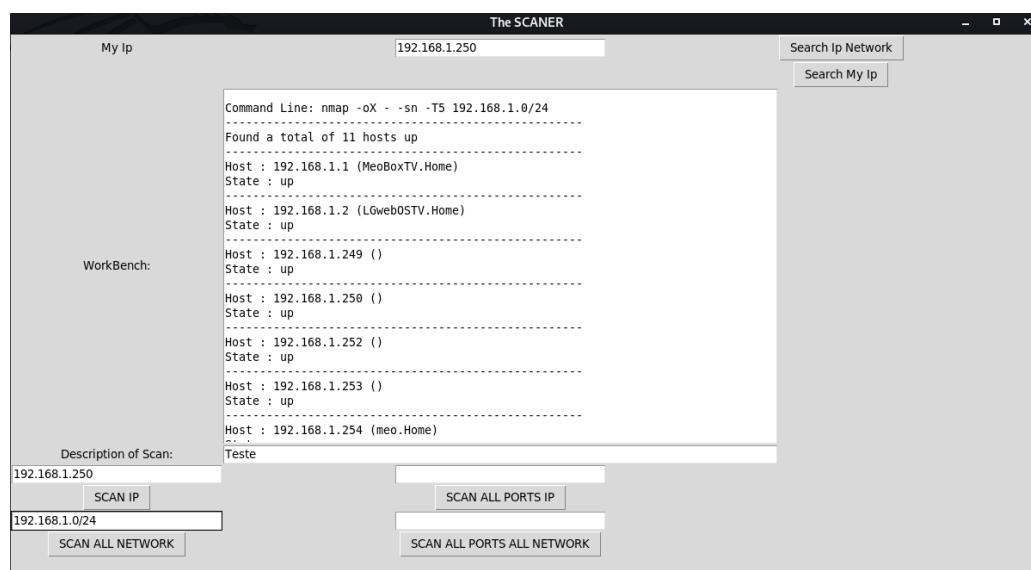


Figura 2-GUI da Ferramenta em python

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.

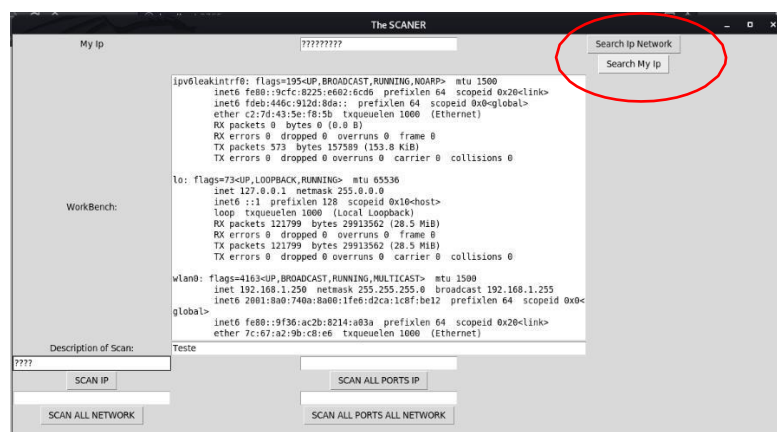


Figura 3-Botões de Ajuda

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.

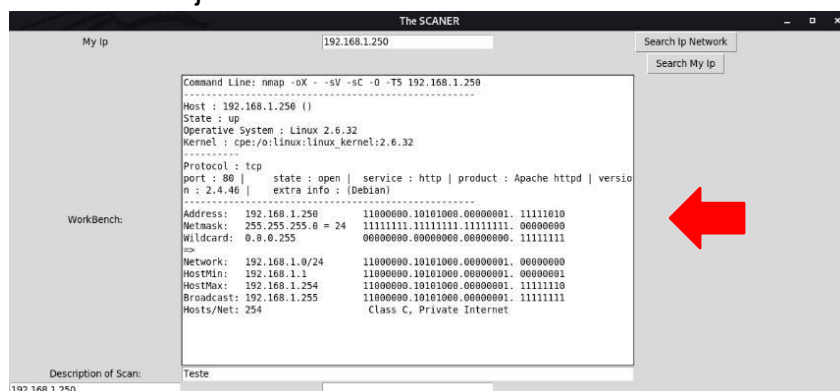


Figura 4-WorkBench / Ecrã Resultados

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.

Edit Scan Delete Scan List Scan	2
Descricao	Teste
Comandline	nmap -oX - -sn -T5 192.168.1.0/24
Id	2
Data	12/14/20
Hora	12/14/20, 12:00 AM
	Command Line: nmap -oX - -sn -T5 192.168.1.0/24

	Found a total of 11 hosts up

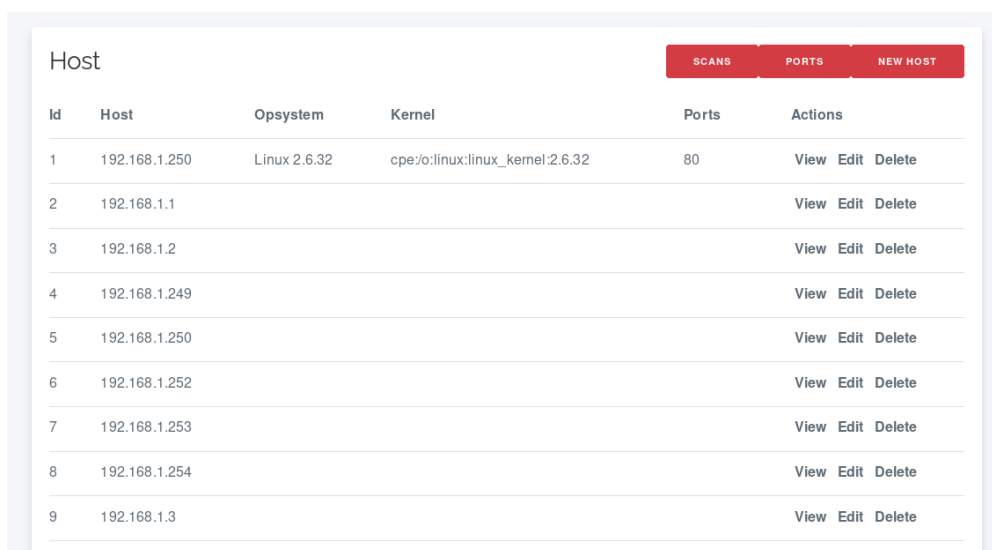
	Host : 192.168.1.1 (MeoBoxTV.Home)
	State : up

	Host : 192.168.1.2 (LGwebOSTV.Home)
	State : up

	Host : 192.168.1.249 ()
	State : up

Figura 5-Scan efetuado e os valores obtidos

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



Host					
			SCANS PORTS NEW HOST		
Id	Host	Opsystem	Kernel	Ports	Actions
1	192.168.1.250	Linux 2.6.32	cpe:/o:linux:linux_kernel:2.6.32	80	View Edit Delete
2	192.168.1.1				View Edit Delete
3	192.168.1.2				View Edit Delete
4	192.168.1.249				View Edit Delete
5	192.168.1.250				View Edit Delete
6	192.168.1.252				View Edit Delete
7	192.168.1.253				View Edit Delete
8	192.168.1.254				View Edit Delete
9	192.168.1.3				View Edit Delete

Figura 6-Vários Scans a varios hosts

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 Scanner - que é logo lançada ao correr o script inicial.

Para lançar o The Scanner 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 contínuo 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 todas as queries SQL. É usado um mysql.connector e liga-se à base de dados do programa. As queries 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.

O objetivo era usar chaves estrangeiras e conseguir ligar cada host e port ao respetivo scan, mas não foi possível desenvolver uma solução às dificuldades encontradas com os instrutores da formação.

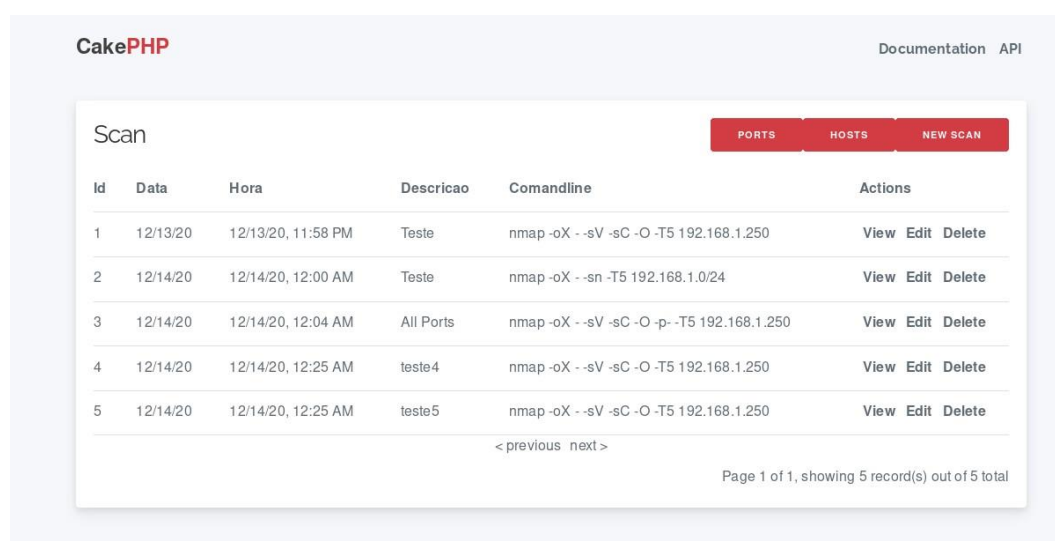
3- Resultados Cake PHP

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:



CakePHP Documentation API

Scan

PORTS HOSTS NEW SCAN

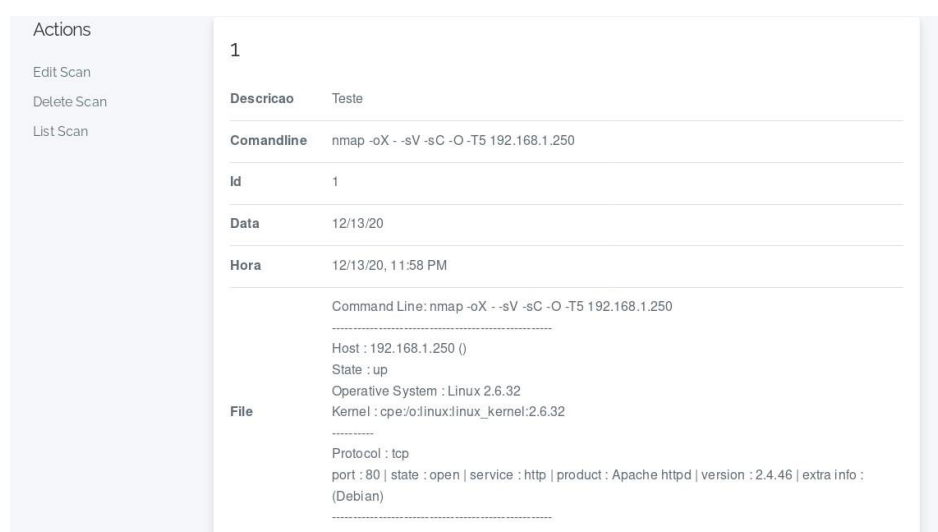
Id	Data	Hora	Descricao	Comandline	Actions
1	12/13/20	12/13/20, 11:58 PM	Teste	nmap -oX - -sV -sC -O -T5 192.168.1.250	View Edit Delete
2	12/14/20	12/14/20, 12:00 AM	Teste	nmap -oX - -sn -T5 192.168.1.0/24	View Edit Delete
3	12/14/20	12/14/20, 12:04 AM	All Ports	nmap -oX - -sV -sC -O -p- -T5 192.168.1.250	View Edit Delete
4	12/14/20	12/14/20, 12:25 AM	teste4	nmap -oX - -sV -sC -O -T5 192.168.1.250	View Edit Delete
5	12/14/20	12/14/20, 12:25 AM	teste5	nmap -oX - -sV -sC -O -T5 192.168.1.250	View Edit Delete

< previous next >

Page 1 of 1, showing 5 record(s) out of 5 total

Figura 7-Visualização dos varios scans efetuados no cake php

e ver cada scan individualmente:



Actions	1
Edit Scan	Descricao Teste
Delete Scan	Comandline nmap -oX - -sV -sC -O -T5 192.168.1.250
List Scan	Id 1
	Data 12/13/20
	Hora 12/13/20, 11:58 PM
	Command Line: nmap -oX - -sV -sC -O -T5 192.168.1.250
	Host : 192.168.1.250 ()
	State : up
	Operative System : Linux 2.6.32
	Kernel : cpe:/o:linux:linux_kernel:2.6.32
	File
	Protocol : tcp
	port : 80 state : open service : http product : Apache httpd version : 2.4.46 extra info : (Debian)

Figura 8-Visualização em detalhe de um scan

Actions
Edit Scan
Delete Scan
List Scan

3

Descricao

All Ports

Comandline

nmap -oX - -sV -sC -O -p- -T5 192.168.1.250

Id

3

Data

12/14/20

Hora

12/14/20, 12:04 AM

File

Command Line: nmap -oX - -sV -sC -O -p- -T5 192.168.1.250

Host : 192.168.1.250 ()

State : up

Operative System : Linux 2.6.32

Kernel : cpe:/o:linux:linux_kernel:2.6.32

Protocol : tcp

port : 80 | state : open | service : http | product : Apache httpd | version : 2.4.46 | extra info : (Debian)

port : 1716 | state : open | service : xmsg | product : | version : | extra info :

Figura 9-Visualização em detalhe de um scan

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

Permite ver os hosts e as portas descobertas:

CakePHP

Documentation API

Port

SCANS HOSTS NEW PORT

Id	Number	State	Service	Version	Info	Actions
1	80	open	http	Apache httpd	2.4.46	View Edit Delete
2	80	open	http	Apache httpd	2.4.46	View Edit Delete
3	1716	open	xmsg			View Edit Delete

< previous next >

Page 1 of 1, showing 3 record(s) out of 3 total

Host

SCANS PORTS NEW HOST

Id	Host	Opsystem	Kernel	Ports	Actions
1	192.168.1.250	Linux 2.6.32	cpe:/o:linux:linux_kernel:2.6.32	80	View Edit Delete
2	192.168.1.1				View Edit Delete
3	192.168.1.2				View Edit Delete
4	192.168.1.249				View Edit Delete
5	192.168.1.250				View Edit Delete
6	192.168.1.252				View Edit Delete
7	192.168.1.253				View Edit Delete
8	192.168.1.254				View Edit Delete
9	192.168.1.3				View Edit Delete

Figura 10-Visualização de Ports e Hosts no cake php (respetivamente)

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 noutra 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_installation, 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 às instalações e configurações que tem de fazer.

Versão 2 (Correções ao código e ao relatório)

Corrigido Introdução, Conclusão e estrutura do relatório

Modificações na introdução, nomeadamente na introdução ao tema – porquê e o quê.
Modificações na conclusão, realçando os pontos fortes e fracos que encontrei junto com uma pequena autoavaliação da totalidade do trabalho e relatório.
Modificações na estrutura como por exemplo a adição de legendas às imagens, entre outros.

Adicionado secção Requisitos Mínimos e link Github

Adicionada a secção de Requisitos Mínimos a cumprir para a utilização da ferramenta e o respectivo link do github.

Corrigido ACE (Arbitrary code execution)

É usado a biblioteca de ipaddress que levanta uma exceção se o input não for um IP
try:

```
ipaddress.ip_network(network)
```

except:

```
messagebox.showerror('Enter your ip!', 'Enter a correct IP!')
```

```
return
```

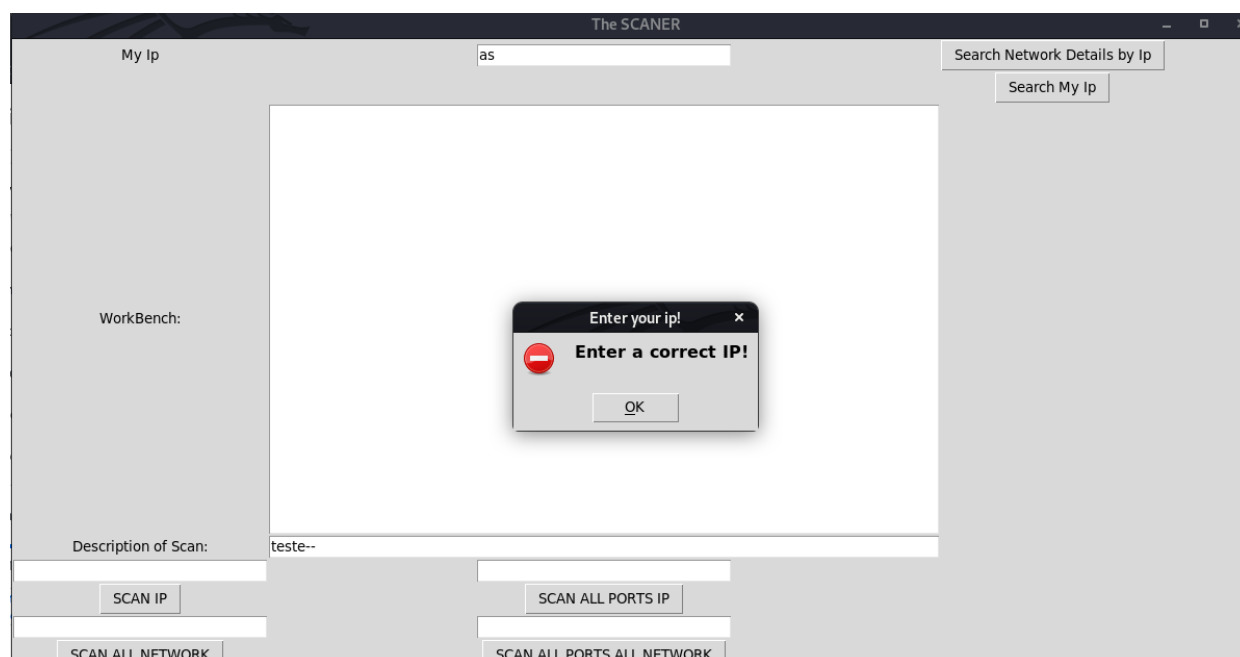


Figura 11 - Mensagem de erro ao não apresentar um ip

Corrigido SQL Injection

É feito um controle muito restrito ao input do utilizador, quer em caracteres nocivos para as queries sql quer em tamanho.

```
def protect_Description(desc):
```

```
    bad_chars = [',', '\\', '"', '"-']
```

```
escape=["\\", '\\', '\\n', '\\r', '\\t', '\\b', '\\f']
size=15
if len(desc) > size:
    messagebox.showerror('Description Error', 'Choose a description 15 characters!')
    return False
for char in bad_chars:
    if desc.find(char)>0:
        messagebox.showerror('Description Error', 'Remove strange characters please!')
        return False
for char in escape:
    if desc.find(char)>0:
        messagebox.showerror('Description Error', 'Remove strange characters please!')
        return False
return True
```

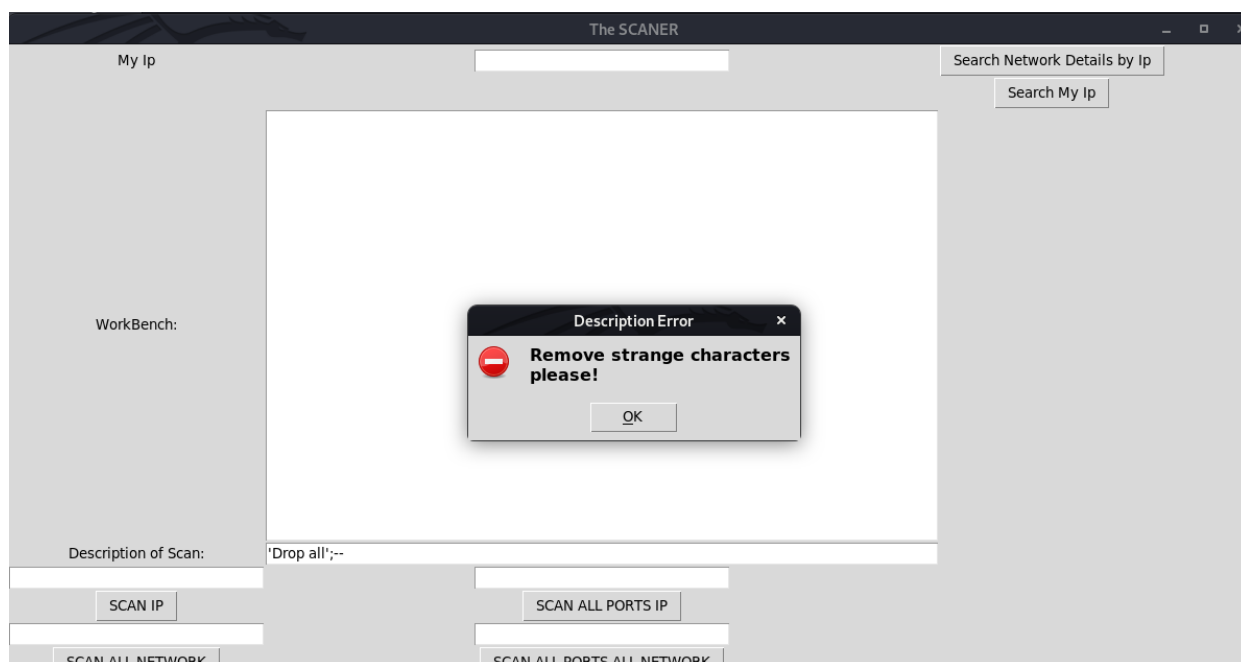


Figura 12 - Mensagem de erro por mau input no campo Descrição

Corrigido instalação de Apache desnecessária

Retirado da instalação do FastScriptInstallation.sh

Conclusão

Ao analisar todo o trabalho e esforço dedicados á elaboração da ferramenta, consigo destacar os seguintes pontos fortes:

- Facilidade de uso
- Variedade de tecnologias usadas
- Apresentação ao utilizador

Consigo também encontrar pontos fracos e de melhoria:

- Não suporta multiplataforma
- Estética da ferramenta

Penso que a ferramenta contém uma boa parte dos temas abordados na Academia.

Anexo

Screenshots

CakePHP Documentation API

Scan

PORTS
HOSTS
NEW SCAN

Id	Data	Hora	Descricao	Comandline	Actions
1	12/13/20	12/13/20, 11:58 PM	Teste	nmap -oX - -sV -sC -O -T5 192.168.1.250	View Edit Delete
2	12/14/20	12/14/20, 12:00 AM	Teste	nmap -oX - -sn -T5 192.168.1.0/24	View Edit Delete
3	12/14/20	12/14/20, 12:04 AM	All Ports	nmap -oX - -sV -sC -O -p- -T5 192.168.1.250	View Edit Delete
4	12/14/20	12/14/20, 12:25 AM	teste4	nmap -oX - -sV -sC -O -T5 192.168.1.250	View Edit Delete
5	12/14/20	12/14/20, 12:25 AM	teste5	nmap -oX - -sV -sC -O -T5 192.168.1.250	View Edit Delete

< previous next >

Page 1 of 1, showing 5 record(s) out of 5 total

The SCANNER

My Ip: 192.168.1.250

WorkBench:

Command Line: nmap -oX - -sV -sC -O -T5 192.168.1.250

Host : 192.168.1.250 ()

State : up

Operative System : Linux 2.6.32

Kernel : cpe:/o:linux:linux_kernel:2.6.32

Protocol : tcp

port : 80	state : open	service : http	product : Apache httpd versio
n : 2.4.46	extra info : (Debian)		

Address: 192.168.1.250 11000000.10101000.00000001.11111010

Netmask: 255.255.255.0 = 24 11111111.11111111.11111111.00000000

Wildcard: 0.0.0.255 00000000.00000000.00000000.11111111

Network: 192.168.1.0/24 11000000.10101000.00000001.00000000

HostMin: 192.168.1.1 11000000.10101000.00000001.00000001

HostMax: 192.168.1.254 11000000.10101000.00000001.11111110

Broadcast: 192.168.1.255 11000000.10101000.00000001.11111111

Hosts/Net: 254 Class C, Private Internet

Search Ip Network

Search My Ip

Description of Scan: Teste

192.168.1.250

SCAN IP

SCAN ALL PORTS IP

SCAN ALL NETWORK

SCAN ALL PORTS ALL NETWORK

The SCANNER

My Ip: 192.168.1.250

WorkBench:

Command Line: nmap -oX - -sn -T5 192.168.1.0/24

Found a total of 11 hosts up

Host : 192.168.1.1 (NeoBoxTV.Home)

State : up

Host : 192.168.1.2 (LOWebOSTV.Home)

State : up

Host : 192.168.1.249 ()

State : up

Host : 192.168.1.250 ()

State : up

Host : 192.168.1.252 ()

State : up

Host : 192.168.1.253 ()

State : up

Host : 192.168.1.254 (mco.Home)

Search Ip Network

Search My Ip

Description of Scan: Teste

192.168.1.250

SCAN IP

SCAN ALL PORTS IP

192.168.1.0/24

SCAN ALL NETWORK

SCAN ALL PORTS ALL NETWORK

Ficheiro de criação da base de dados

```
1- USE cakephp;
2- CREATE TABLE `host` (
3-   `id` int(11) NOT NULL AUTO_INCREMENT,
4-   `host` varchar(100) NOT NULL,
5-   `opsystem` varchar(100) DEFAULT 'Unknow',
6-   `kernel` varchar(100) DEFAULT 'Unknow',
7-   `ports` varchar(500) DEFAULT 'Unknow',
8-   PRIMARY KEY (`id`)
9- ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
10-
11-
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;
```

Ficheiro de instalação python

```
#!/bin/bash
#Python
apt-get install python
apt-get install pip
pip install python-nmap
pip install mysql-connector-python
pip install ipaddress
apt-get install ipcalc
apt-get install nmap
```

FastInstalation

```
#!/bin/bash
echo '====='
echo 'Fast Instalation, run as sudo'
echo '====='
echo 'Install php'
echo '====='
apt install php libapache2-mod-php php-common php-gmp php-curl php-intl
php-mbstring php-xmlrpc php-mysql php-gd php-imap php-ldap php-cas php-
bcmath php-xml php-cli php-zip php-sqlite3
echo 'Install composer'
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'756890a4488ce9024fc62c56153228907f1545c228516cbf63f885e036d37e9a59d27d63
f46af1d4d07ee0f76181c7d3') { echo 'Installer verified'; } else { echo
'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
mv composer.phar /usr/local/bin/composer
echo '====='
echo 'Instaling mariaDb'
echo '====='
apt install mariadb-server mariadb-client
echo '====='
echo 'Configure Database'
echo 'Choose a root pasword and choose YES on all'
echo '====='
mysql_secure_installation
while :
do
echo '====='
echo 'Type again the root password you choosed'
echo '====='
read -p 'Password:' password
echo Password was $password ?
echo 'Yes=1 or No=2'
read conf
if [ $conf -eq 1 ]
then
break
fi
done
echo '====='
echo 'Starting Database Server'
echo '====='
systemctl start mariadb.service
systemctl enable mariadb.service
echo '====='
echo 'Configuring Database'
echo '====='
mysql -u root --password=$password -Bse "DROP DATABASE IF EXISTS
cakephp;CREATE DATABASE cakephp;DROP USER IF EXISTS
'cakephpuser'@'localhost';CREATE USER 'cakephpuser'@'localhost'
IDENTIFIED BY 'c@k3_Us3r_p@ssw0rd';GRANT ALL ON cakephp.* TO
'cakephpuser'@'localhost' WITH GRANT OPTION;FLUSH PRIVILEGES;EXIT;"
```

```
mysql -u root --password=$password < ./database_without_foreignkeys.sql
#mysql -u root --password=$password < ./setup_database.sql
systemctl start mariadb
echo '=====
echo 'Installing adminer for easier manipulation of database'
echo '=====
apt install adminer
a2enconf adminer.conf
echo '=====
echo 'Adminer in: https://127.0.0.1/adminer/'
echo '=====
echo '=====
echo 'Installing Python Dependencies'
echo '=====
./FastPython.sh
echo '=====
echo 'Creating project'
echo '=====
mkdir /var/www/cakephp
#Compress
#O FICHEIRO PYTHON TEM DE ESTAR NA DIRETORIA DO WWW ROOT DO CAKE - cria
diretoria scans na webroot
#cp scanner.py scansDefinitions.py sqlConnector.py
/var/www/cakephp/app/webroot
#tar -czvf Samuel_ProjectoFinalAnciber.tar.gz ./cakephp
tar -xzvf Samuel_ProjectoFinalAnciber.tar.gz -C /var/www/
#mete o script acessivel de qualquer lado
chmod +x /var/www/cakephp/app/webroot/scanner.py
ln /var/www/cakephp/app/webroot/scanner.py /usr/local/bin/scanner.py
#cd /var/www/cakephp
#composer create-project --prefer-dist cakephp/app
#cd /var/www/cakephp/app/
#composer install && composer update && composer dump-autoload --optimize
#nano /var/www/cakephp/app/config/app_local.php
#Copiar o meu trabalho para aqui
echo '=====
echo 'Checking some last permissions'
echo '=====
chown -R www-data:www-data /var/www/cakephp
chmod -R 755 /var/www/cakephp
echo '=====
echo 'Deploying server'
echo '=====
python3 /var/www/cakephp/app/webroot/scanner.py &
cd /var/www/cakephp/app/bin/
./cake server
echo 'Visit http://localhost:8765/'
```