-- Trabalho I -- CI301 - Segurança Computacional I -- Relatório

Trabalho realizado por:
Luiz Adolpho Baroni GRR20123972
Reginaldo dos Santos Junior GRR20120742

Implementação em PYTHON de um Port Scanner:

O programa todo é muito simples. Faz-se o parser da entrada, em seguida é feito um ping(ip) no ip desejado para saber se está de pé ou não, e então se executa o porttry(ip,port), principal função do programa que abre um socket, usa esse scoket para se conectar numa tupla (ip,porta) , envia uma requisição para receber o cabeçalho da aplicação (banner) que estiver naquela porta, armazena numa variavel e fecha a conexão.
Tentamos capturar o banner sem fazer a requisição, mas algumas aplicações não enviam simplesmente após a conexão, foi necessário enviar um pacote no socket para que o Host respondesse com o banner.

```
1-  def porttry(ip, port): #abre o socket, tenta conectar, pega o banner, fecha e retorna o banner
2-      s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #socket.AF_INET, socket.SOCK_STREAM
3-      s.settimeout(0.5) #timeout para nao esperar para sempre
4-      s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1) #esta opcao permite reutilizar o mesmo socket num curto intervalo de tempo
5-      try:
6-              s.connect((ip, port)) #conecta no ip , usando a porta especifica
7-              s.send("HEAD / HTTP/1.0\r\n\r\n") #envia requisicao do header da aplicacao naquela porta
8-              banner = s.recv(1024)
9-              s.close()
10-             return True, str(banner)
11-     except:
12-             return None, str(0)
```

A linha 7 é a responsável por enviar a requisição "HEAD / HTTP/1.0", seguida de uma linha em branco "\r\n\r\n", que nos retorna apenas o header da aplicação.

A parte mais complicada foi o parser de entrada, que demandou uma certa criatividade e muitas linhas de codigo.

Abaixo o resultado de duas varreduras:

Varredura 1:
    No IP passado em sala (200.238.144.29) com range de 1 a 3000:

```
baroni@baroniPC ~/portScanner> python meuPS.py 200.238.144.29 1-3000
Scanning...
-----------------------------------
HOST:  200.238.144.29  is Up!
--Port 21 opened
--Service:  220 Welcome to the ftp service

--Port 22 opened
--Service:  SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1.9

--Port 80 opened
--Service:  HTTP/1.0 200 OK

-----------------------------------
Time to Scan: 19s
```

Varredura 2:
    No localhost(127.0.0.1) com range de 1 a 3000:

```
baroni@baroniPC ~/portScanner> python meuPS.py 127.0.0.1 1-3000
Scanning...
-----------------------------------
HOST:  127.0.0.1  is Up!
--Port 22 opened
--Service:  SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4

--Port 631 opened
--Service:  HTTP/1.0 400 Bad Request
Content-Language: pt_BR
Content-Length: 346
Content-Type: text/html; charset=utf-8
Date: Sat, 14 Apr 2018 16:34:52 GMT
Accept-Encoding: gzip, deflate, identity
Server: CUPS/2.1 IPP/2.1
X-Frame-Options: DENY
Content-Security-Policy: frame-ancestors 'none'


--Port 3000 opened
--Service:  HTTP/1.0 501 Not Implemented

-----------------------------------
Time to Scan: 0s
```