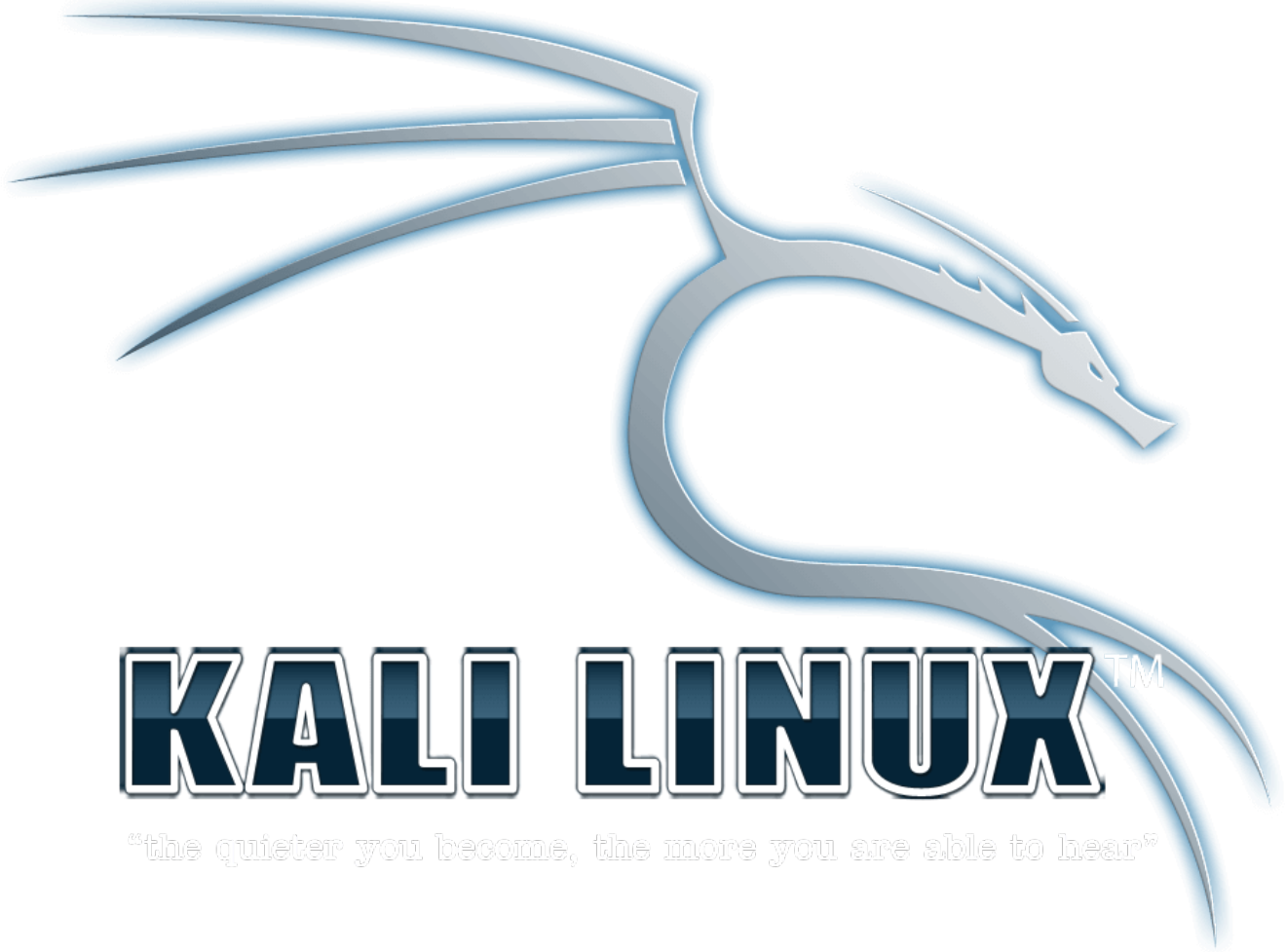
Pentest com Kali Linux



Instrutor:Vitor Mazuco

http://facebook.com/vitormazuco

Email:vitor.mazuco@gmail.com

WebSite:http://vmzsolutions.com.br

Determinando um range de rede

Com a informação recolhida na aula de DNSenum, podemos agora concentrar-se em determinar os endereços IP que vão desde a rede de destino. Agora, vamos explorar as ferramentas necessárias para alcançá-lo.

Determinando um range de rede

Abra uma nova janela de terminal e execute o seguinte comando:

```
# dmitry -wnspb targethost.com -o /root/dmitry-resultado
```

Determinando um range de rede

Quando terminar, deveremos ter um documento de texto com o nome de arquivo *dmitry-resultado.txt*, preenchido com as informações recolhidas a partir do nosso alvo

Determinando um range de rede

Para emitir uma solicitação de máscara de rede ICMP,

digite o seguinte comando:

Usando *scapy*, podemos emitir um *traceroute* paralelo.

Para iniciá-lo, digite o seguinte comando:

```
# scapy
```

Determinando um range de rede

root@kali:~# scapy INFO: Can't import python gnuplot wrapper .

Won't be able to plot. WARNING: No route found for IPv6

destination :: (no default route?) Welcome to Scapy (2.3.2)

>>> ans,unans=sr(IP(dst="www.targethost.com/30", ttl=(1,6))/TCP())

Determinando um range de rede

Para exibir o resultado em uma tabela, emitimos a seguinte função:

```
>>>  ans.make_table( lambda (s,r): (s.dst, s.ttl, r.src) )
```

A saída é mostrado como se segue:

```
216.27.130.162    216.27.130.163
216.27.130.164    216.27.130.165

1  192.168.10.1    192.168.10.1    192.168.10.1    192.168.10.1
2  51.37.219.254   51.37.219.254   51.37.219.254   51.37.219.254
3  223.243.4.254   223.243.4.254   223.243.4.254   223.243.4.254
4  223.243.2.6     223.243.2.6     223.243.2.6     223.243.2.6
5  192.251.254.1   192.251.251.80  192.251.254.1   192.251.251.80
```

Determinando um range de rede

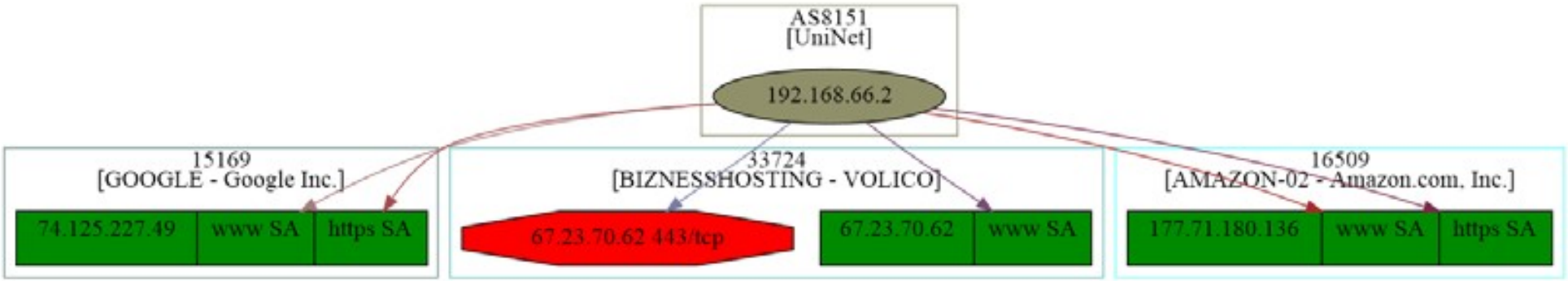
Para obter um *traceroute* TCP com scapy, escreva a seguinte função:

```
>>>  res,unans=traceroute(["www.google.com","www.Kali-linux.org","www.targethost.com"],dport=[80,443],maxttl=20, retry=-2)
```

Para exibir uma representação gráfica do resultado, nós simplesmente emitir a seguinte função:

```
>>> res.graph()
```

Determinando um range de rede



Determinando um range de rede

Para salvar o gráfico, basta digitar a seguinte função:

```
>>> res.graph(target="> /tmp/graph.svg")
```

Para sair scapy, digite a seguinte função:

```
>>> exit()
```

Com os resultados obtidos, podemos agora avançar para documentá-lo.

Determinando um range de rede

Primeiramente, usamos o *dmitry* para obter informações a partir do nosso alvo. A opção *-wnspb* nos permite realizar uma pesquisa de WHOIS sobre o nome de domínio, recuperar as informações com *netcraft.com*, e realizar uma pesquisa para possíveis subdomínios e uma verificação da porta TCP. A opção *-o* nos permite salvar o resultado em um documento de texto. Logo em seguida, fazemos um pedido simples máscara de rede ICMP com a opção *-s* para a saída o endereço IP e de máscara de rede.

Determinando um range de rede

Em seguida, usamos o scapy para emitir um *traceroute multiparallel* no host de destino, exibindo o resultado de uma apresentação em uma tabela. Depois, foi realizado um *traceroute* TCP de várias portas 80 e 443, e definimos o máximo de 20 TTL(limite de 255) para parar o processo. Com o resultado obtido, criamos uma representação gráfica do mesmo, e ele foi salvo em um diretório temporário. Por fim, saímos do scapy.