

Descrição

Foi adicionada a análise do tráfego ipv6, mostrando os campos : version, traffic class: Flow Label, payload length, next header, hop limit,source address, destination address. Para isso foi criada a função ipv6Header.

A função recebe um data que é um array em bytes tamanho de 40 bytes.

```
def ipv6Header(data):
```

Através da função struct unpack descompacta o valor compactado em sua representação original com o formato especificado. Para descompactar o data foi utilizado o formato IHBB, que gera um vetor de 4 posições:

[0] I - unsigned int - 4 bytes [Version, Traffic Class, Flow Label]
[1] H - unsigned short - 2 bytes [Payload Length]
[2] B - unsigned char - 1 byte [Next Header]
[3] B - unsigned char - 1 byte [Hop Limit]

```
packet = struct.unpack(">IHBB", data[0:8])
```

Através da função socket.inet_ntop os endereços são convertidos

```
srcAddress = socket.inet_ntop(socket.AF_INET6, data[8:24])  
dstAddress = socket.inet_ntop(socket.AF_INET6, data[24:40])
```

Para pegar o valor do campo version foi obtido através do shift de 28 bits da primeira posição do array.

```
version = packet[0] >> 28
```

Para pegar o valor do campo traffic class foi obtido através do shift de 16 bits do restante da primeira posição do array e depois convertido para um valor inteiro.

```
traffic_class = packet[0] >> 16  
traffic_class = int(traffic_class) & 4095
```

O valor do campo flow label é o valor resultante dos dois shifts anteriores convertido para inteiro.

```
flow_label = int(packet[0]) & 65535
```

O valor do payload length é equivalente a segunda posição do array.

```
payloadLength = packet[1]
```

O valor do next header é equivalente a terceira posição do array.

```
nextHeader = packet[2]
```

O valor hop limit é equivalente a quarta posição do array.

```
hopLimit = packet[3]
```

Após é feito o print de todos os campos calculados.

```
print "*****IPv6 HEADER*****"
print "\tVersion: " + str(version)
print "\tTraffic class: " + str(hex(traffic_class))
print "\tFlow label: " + str(hex(flow_label))
print "\tPayload Length: " + str(payloadLength)
print "\tNext Header: " + str(nextHeader)
print "\tHop limit: " + str(hopLimit)
print "\tSource Address: " + str(srcAddress)
print "\tDestination Address: " + str(dstAddress)
```

PUBLICAÇÃO

Já que o repositório não era atualizado faz algum tempo enviei um e-mail para o dono do repositório, mas até hoje o e-mail não foi correspondido.

packetAnalyzerAndSniffer project ➤ Caixa de entrada x



Maria Carolina Webber do Prado Lima <MCWPLima@ucs.br>
para demola.hussainin ▾

dom., 14 de jun. 16:04 ☆ ↩ ⋮

Hello my name is maria. I am a computer science student in Brazil. For a college project I used your project as a basis. I implemented ipv6 in the project, since I had only implemented ipv4. Would you like to put this implementation in your project?
Thank you very much in advance.