

Packet Decoder

De module `hva_pckt` bevat routines om Ethernet-, IP- en TCP- packet te decoderen. Deze routines zijn illustratief en behandelen niet alle mogelijke packetten. Ze voldoen om de meeste ethernet-pcaps te lezen, kunnen gebruikt worden om de belangrijkste aspecten van tcp/ip te verduidelijken

hva_pckt.py

MacAddress

Een class om een ethernet-mac-adres van 6-bytes te bevatten.

`__init__`

`regel 8–10` MacAddress bewaart zijn gegevens in 6 bytes netwerk-order.

`regel 11–13` De string notatie (hex gescheiden door :) wordt geconverteerd naar het interne formaat.

`__str__`

`regel 17–19` Het interne formaat wordt geconverteert naar een string.

IpAddress

Een class om een ip-adres van 4-bytes te bevatten.

`__init__`

`regel 23–25` IpAddress bewaard zijn gegevens in 4 bytes netwerk-order.

`regel 26–28` De string notatie (decimal gescheiden door .) wordt geconverteerd naar het interne formaat.

`regel 29–30` Een Ipaddress kan ook geïnitieerd worden door een integer waarde.

`__str__`

`regel 34–36` Het interne formaat wordt geconverteert naar een string.

`__int__`

regel 38–39 Het interne formaat wordt geconverteert naar het int.

__lt__ ... __le__

regel 36–47 Vergelijking operatoren op IpAddress.

IpPort

TODO Het toevoegen van een class `IpPort`

Eth

Een class om een "Ethernet Type II" te representeren. Het type/size veld bevat de waarde `0x0800` .

decode

regel 54–65 We gebruiken een class-method om een instance te creëren en te initialiseren met de meegeven bytes. De header is 14 bytes groot en bevat source-mac-adres (smac), destination-mac-adres (dmac) en het type (typ). De rest van de bytes zijn de payload(pl) van het packet.

Het type zou `0x0800` moeten zijn, indien niet return `None` .

__str__

regel 67–68 Voor het gemak een string-represenatie met alle non-payload-velden, en de lengte van de payload.

Ip:

Een class om een "IP versie 4" te representeren. De versie moet 4 bevatten.

decode

regel 72–94 We gebruiken een class-method om een instance te creëren en te initialiseren met de meegeven bytes. De header is 20 bytes groot en bevat o.a. protocol (proto), status-vlaggen (flgs), source-ip-adres (saddr), destination-adres (daddr) optionele opties (opts). De rest van de bytes zijn de payload(pl) van het packet. Het versie zou `4` moeten zijn, indien niet return `None` .

__str__

regel 95–101 Voor het gemak een string-represenatie met alle non-payload-velden, en de lengte van de payload.

TCP:

Tcp verzorgt een betrouwbare verbinding tussen twee end-points. Het heeft alleen zijn een packet als TCP te coderen als in het IP-packet `proto == 6` (TCP).

decode

`regel 105-122` We gebruiken een class-method om een instance te creëren en te initialiseren met de meegeven bytes. De header is 20 bytes groot en bevat o.a. source port (sport), destination port (dport), sequence id (seq), acknowledgement id (ack), optionele opties (opts). De rest van de bytes zijn de payload(pl) van het packet.

__str__

`regel 124-129` Voor het gemak een string-represenatie met alle non-payload-velden, en de lengte van de payload.

Udp

TODO Het toevoegen van een class `Udp`