Präsentation von Luca Chmielarski
WWI21SEB

Agenda

1. Was ist Scapy?
2. Installation
3. Grundlagen
4. Live-Demo

> https://github.com/930C/scapy <

Scapy

- Interaktive Python Library zur Manipulation von Pakets
- Hauptmerkmale:
  - Generieren und Versenden von eigenen Pakets
  - Manipulation von Pakets (Header & Payload)
  - Netzwerkanalyse

https://scapy.net/

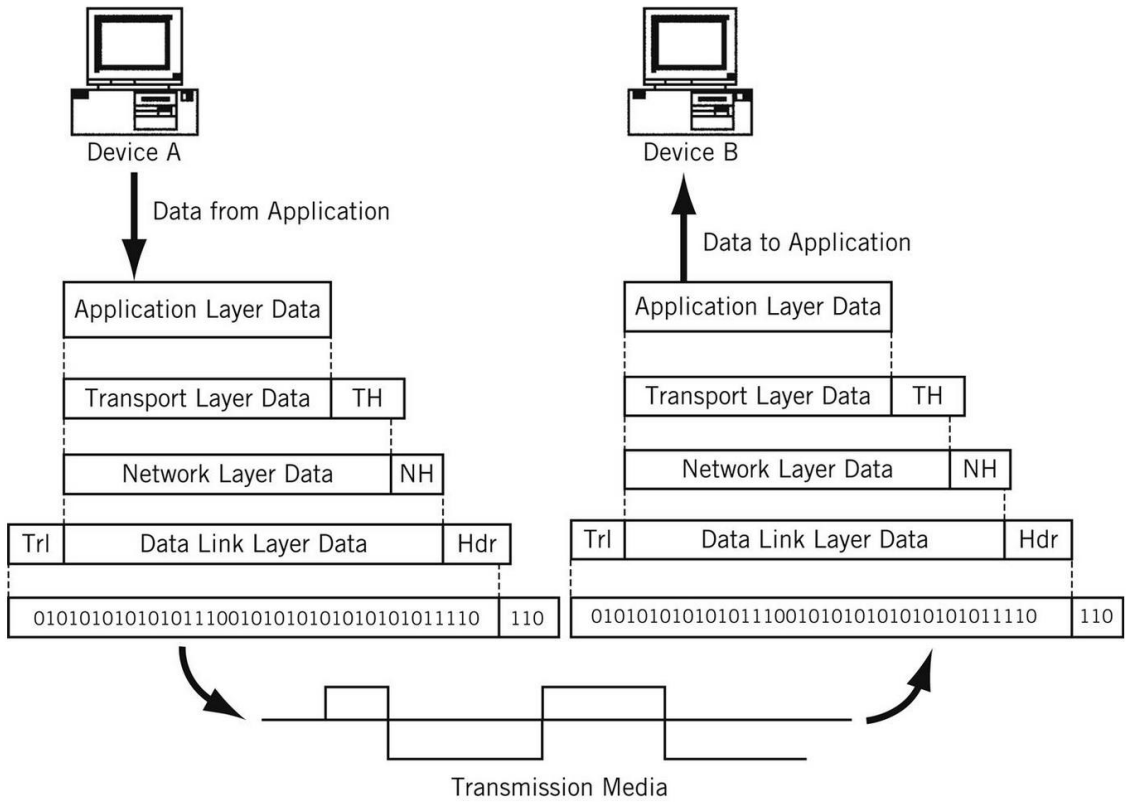
https://scapy.readthedocs.io/en/latest/

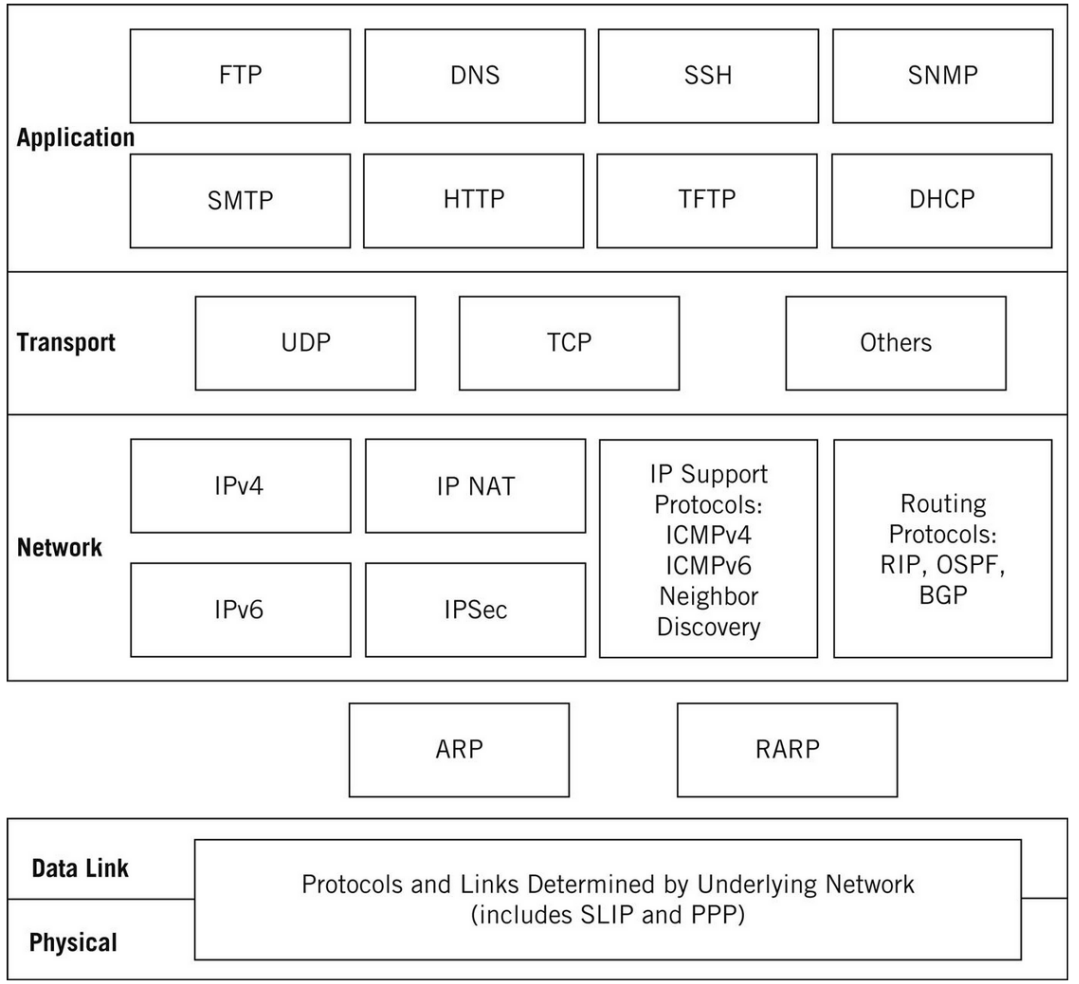
Installation

https://scapy.readthedocs.io/en/latest/installation.html#

Grundlagen

TCP/IP Referenz Modell





Quelle: [The Illustrated Network, 2nd Edition – Morgan Kaufmann (O'reilly)]

Grundlagen

Domain Specific Language

TCP/IP-Schicht	Protokoll	Scapy-Objekt
Link Layer	Ethernet	`Ether()`
	Wi-Fi (802.11)	`Dot11()`
	ARP	`ARP()`
Internet Layer	IP	`IP()`
	IPv6	`IPv6()`
	ICMP	`ICMP()`
Transport Layer	TCP	`TCP()`
	UDP	`UDP()`
	SCTP	`SCTP()`
Application Layer	HTTP	`Raw()` (mit `IP()` und `TCP()`)
	DNS	`DNS()`, `DNSQR()`, `DNSRR()`
	FTP	`Raw()` (mit `IP()` und `TCP()`)
	SMTP	`Raw()` (mit `IP()` und `TCP()`)

Methode	Beschreibung
`send()`	Sendet Pakete auf Layer 3 (Netzwerkschicht). Beispiel: `send(IP(dst="1.1.1.1")/ICMP())`
`sendp()`	Sendet Pakete auf Layer 2 (Datenverbindungsschicht). Beispiel: `sendp(Ether()/IP(dst="1.1.1.1")/ICMP())`
`sr()`	Sendet und empfängt Pakete auf Layer 3. Beispiel: `sr(IP(dst="1.1.1.1")/ICMP())`
`sr1()`	Sendet ein Paket und empfängt die erste Antwort auf Layer 3. Beispiel: `sr1(IP(dst="1.1.1.1")/ICMP())`
`srp()`	Sendet und empfängt Pakete auf Layer 2. Beispiel: `srp(Ether()/IP(dst="1.1.1.1")/ICMP())`
`srp1()`	Sendet ein Paket und empfängt die erste Antwort auf Layer 2. Beispiel: `srp1(Ether()/IP(dst="1.1.1.1")/ICMP())`
`sniff()`	Fängt Pakete ab und kann mit einem Filter und einer Callback-Funktion verwendet werden. Beispiel: `sniff(filter="icmp", prn=lambda x: x.show())`
`traceroute()`	Führt eine Traceroute durch und zeigt den Weg zu einem Ziel. Beispiel: `traceroute("1.1.1.1")`
`arping()`	Sendet ARP-Anfragen, um Geräte im Netzwerk zu entdecken. Beispiel: `arping("192.168.1.0/24")`
`sendrecv()`	Eine generische Methode, die `sr()`, `sr1()`, `srp()`, und `srp1()` vereint und basierend auf den Parametern die passende Methode auswählt. Beispiel: `sendrecv(IP(dst="1.1.1.1")/ICMP())`

```
>>> p = Ether() / IP(dst="test.c930.net") / TCP(flags="F")
>>> p.summary()
'Ether / IP / TCP 10.50.78.143:ftp_data > Net("test.c930.net/32"):http F'
```