Prontuario di comandi connessi a network security

- Per lo sniffing (intercettazione e analisi del traffico)
  - tcpdump
  - wireshark
- Per la gestione dei certificati TLS
  - openssl

tcpdump / pcap

- tcpdump è un tool a riga di comando
  - output testuale, direttamente visibile in tempo reale anche in remoto via ssh
  - output *raw* in formato pcap, rileggibile da tcpdump stesso o Wireshark per analisi successiva
- i *capture filters* permettono di selezionare i pacchetti interessanti
  - sintassi comune per tcpdump e Wireshark, dipende dalla libreria comune libpcap
  - **`man pcap-filter`** (7)
- i *display filters* permettono di limitare i pacchetti visualizzati o di cambiare il formado dell'output
  - per Wireshark, soprattutto interfaccia grafica
  - per tcpdump, opzioni del comando

capture filters comuni

- selezione per ip origine/destinazione

  `[dst|src] [ip|arp] host <ip o nome>`

- selezione per porta origine/destinazione

  `[tcp|udp] [dst|src] port <porta>`

- selezione per protocollo

  `ip proto <tcp|udp|icmp|ah|esp|...>`

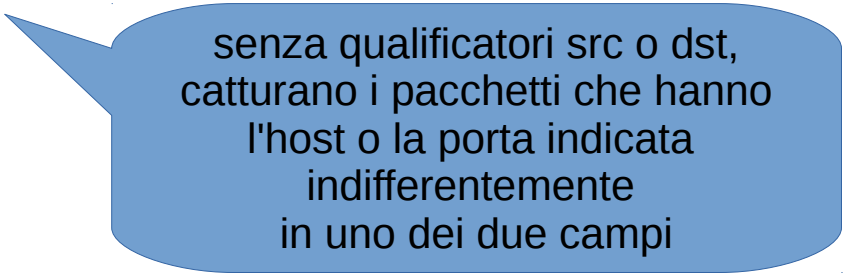
- combinazione in espressioni logiche

  `and or not ()`

- Esempi

  `ip proto tcp and dst port 80`

  `src host 10.1.1.1 and ( tcp port 22 or udp port 53 )`



tcpdump

- le opzioni più comuni di tcpdump

`-i interface` (any=tutte)

`-w outputfile` (output raw)

`-v` (verbose – attenzione può produrre più righe per ogni pacchetto!)

`-n` (non converte indirizzi e porte in nomi)

`-l` (line buffered output)

`-p` (no promiscuous mode)

`-s size` (byte catturati per pacchetto, default 256k)

`-c count` (esce dopo count pacchetti catturati)

`-A` (stampa i pacchetti in ASCII)

`-X` (stampa i pacchetti in ASCII ed esadecimale)

Wireshark

- Il modo più sicuro di usarlo è senza privilegi di root, ma deve essere abilitato:
  - `sudo dpkg-reconfigure wireshark-common`

    (rispondere che si vuole consentire cattura come non-root)
  - `adduser <utente> wireshark`

    (sulle VM non serve, las è gia membro del gruppo)
- Richiede interfaccia grafica! Per usarlo direttamente in remoto:
  - X supporta client (applicazioni) e server (gestore di mouse-tastiera-scheda video) su host diversi
  - usa porte normalmente non aperte, difficili da ricordare, ecc...
  - ssh può fare tunneling del protocollo X!

    `ssh -X las@192.168.56.X wireshark`
- Utilizzo in combinazione con tcpdump -w file.pcap
  - facile sniffing remoto, da riga di comando o anche schedulato
    - file pcap trasferito su workstation di gestione (non disponibile sugli host del laboratorio), caricato su `wireshark -r file.pcap` per analisi "friendly"
    - anche in pipeline

      `ssh root@192.168.56.201 tcpdump -i any -w - | wireshark -i -`

openssl

- Suggerimento: verificare quanto prodotto dalla mini-CA di OpenVPN
  - dopo build-ca
    - visualizzazione della chiave generate per la CA

      `openssl rsa -in /etc/openvpn/easy-rsa/keys/ca.key -text -noout`
    - visualizzazione del certificato della CA

      `openssl x509 -in /etc/openvpn/easy-rsa/keys/ca.crt -text -noout`
  - dopo build-key-server

    stesse verifiche con `server.key` e `server.crt`
- Di uso comune:
  - passaggi per la certificazione di una chiave:
    - generazione di una nuova coppia di chiavi e della CSR

      `openssl req -nodes -newkey rsa:2048 -keyout example.key -out example.csr`
    - (svolto dalla CA) trasformazione della CSR in un certificato

      `openssl x509 -req -in example.csr -days 365 -CA ca.crt -CAkey ca.key -set_serial 01 -out example.crt`
  - generazione di una chiave auto-firmata

    `openssl req -nodes -newkey rsa:2048 -keyout example.key -out example.crt -x509 -days 365`
  - verifica che un certificato corrisponda a una chiave privata e a una CSR (match fingerprint del modulo)

    `openssl rsa -noout -modulus -in example.key | openssl sha256`

    `openssl x509 -noout -modulus -in example.crt | openssl sha256`

    `openssl req -noout -modulus -in example.csr | openssl sha256`
  - openssl come client TLS a riga di comando, con stampa della catena dei certificati

    `openssl s_client -showcerts -host example.com -port 443 </dev/null`