

# Port scans detector with prolog

Andrea Imparato - Lorenzo Tessari

Università degli studi di Padova

September 20, 2011

Software che riconosce la presenza di un port scanning su di un host in rete.

**Port scanning** = enumerazione porte attive/filtrate/chiusse

2 tipologie di scanning:

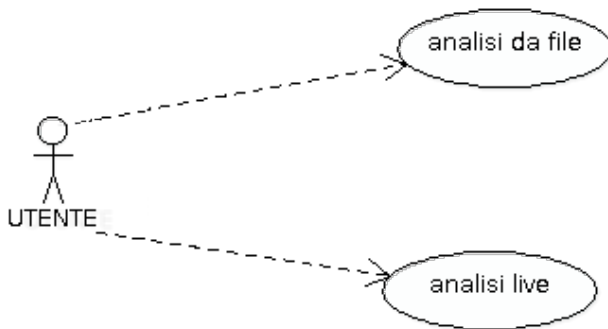
- tcp scan
- syn scan

- Tema sicurezza: conoscenza personale tool di sicurezza come **nmap** (port scanning) e **wireshark** (sniffing)

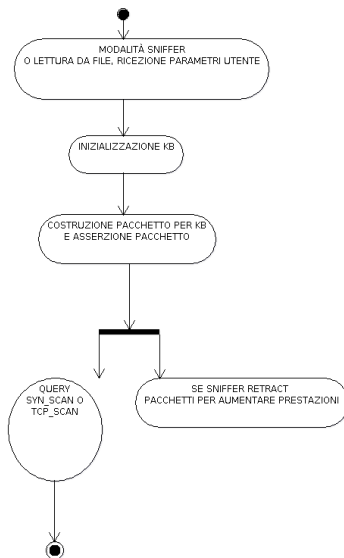
- Tema sicurezza: conoscenza personale tool di sicurezza come **nmap** (port scanning) e **wireshark** (sniffing)
- Snort IDS modulo port scanning non adeguato

- Tema sicurezza: conoscenza personale tool di sicurezza come **nmap** (port scanning) e **wireshark** (sniffing)
- Snort IDS modulo port scanning non adeguato
- Prolog!

# Use case



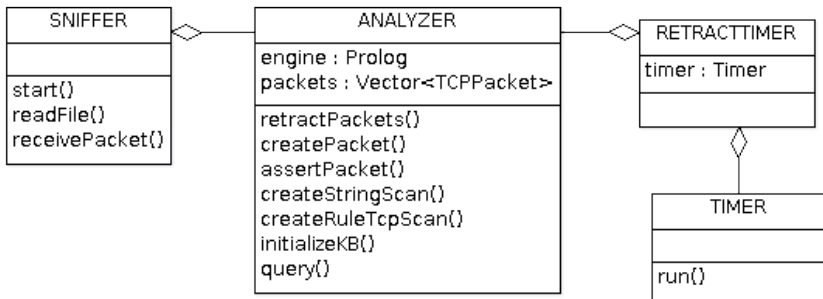
# Diagramma delle attività



# Progettazione

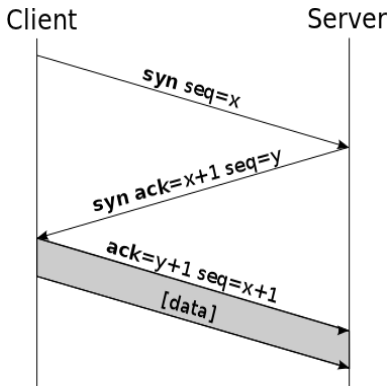
Librerie:

- jpcap  $\Rightarrow$  sniffer
- tuprolog  $\Rightarrow$  KB





# Protocollo Tcp/Ip



# Inferenza - Riconoscimento connessione tcp

```
/* regola per connessione tcp */  
connessione_tcp(SOURCE, DESTINATION, SP, DP):—  
pacchetto(SP, DP, syn, SOURCE, DESTINATION, X, 0)  
, pacchetto(DP, SP, syn, DESTINATION, SOURCE, Y, Z)  
, pacchetto(SP, DP, SOURCE, DESTINATION, Z, W)  
, Z is X+1, W is Y+1.
```

```
/* regola per connessione connessione syn */  
connessione_syn(SOURCE, DESTINATION, SP, DP): –  
pacchetto(SP, DP, syn, SOURCE, DESTINATION, X, 0)  
, pacchetto(DP, SP, syn, DESTINATION, SOURCE, Y, Z)  
, Z is X+1.
```

# Inferenza - Riconoscimento porta chiusa

```
/* regola per riconoscere se la porta e' chiusa */  
porta_chiusa(SOURCE, DESTINATION, SP, DP):—  
pacchetto(SP, DP, syn, SOURCE, DESTINATION, X, 0)  
, pacchetto(DP, SP, rst, DESTINATION, SOURCE, 0, Z)  
, Z is X+1.
```

# Inferenza scan 4 porte

```
tcp_scan(X,Y):-  
  connessione_tcp(X,Y,A1,A2),A1\=A2,  
  connessione_tcp(X,Y,A3,A4),  
  A3\=A4, connessione_tcp(X,Y,A5,A6)  
  ,A5\=A6, connessione_tcp(X,Y,A7,A8)  
  ,A7\=A8,A2\=A4,A2\=A6,A2\=A8,A4\=A6  
  ,A4\=A8,A6\=A8.
```

```
syn_scan(X,Y):-  
  connessione_syn(X,Y,A1,A2)  
  ,A1\=A2, connessione_syn(X,Y,A3,A4),A3\=A4  
  , connessione_syn(X,Y,A5,A6),A5\=A6,  
  connessione_syn(X,Y,A7,A8),A7\=A8,A2\=A4,A2\=A6,  
  A2\=A8,A4\=A6,A4\=A8,A6\=A8.
```

- In presenza di poco traffico sulla rete prestazioni ottime. Zero risultati di falsi positivi/negativi sia per tcp scan sia syn scan.

- In presenza di poco traffico sulla rete prestazioni ottime. Zero risultati di falsi positivi/negativi sia per tcp scan sia syn scan.
- Syn scan difficile da trovare con molto traffico. Euristiche nmap?

- In presenza di poco traffico sulla rete prestazioni ottime. Zero risultati di falsi positivi/negativi sia per tcp scan sia syn scan.
- Syn scan difficile da trovare con molto traffico. Euristiche nmap?
- Tcp scan quasi banale anche con molto traffico.



# DEMO!