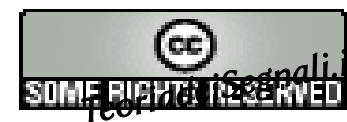




**INFO-COM Dpt.**  
**Dipartimento di Scienza e Tecnica**  
**dell'Informazione e della Comunicazione**  
**Università degli Studi di Roma La Sapienza**



# Strumenti Linux per la sicurezza wireless



# Contenuti

Approccio della esposizione

Sicurezza del proprio computer

- Qualità e recupero delle password

- Detezione dell'intrusione e controllo dell'integrità del sistema

- Firewall di Linux e sua configurazione

- Scansione delle porte in ascolto

Sicurezza della propria rete

- Scansione delle vulnerabilità

- Sniffing e Ispezione profonda dei pacchetti

- Detezione e prevenzione delle intrusioni

- Antispam e antivirus

Sicurezza dell'accesso radio

- Architettura e protocollo, sniffing wireless, mappatura

- Violazione della riservatezza, intrusione nella rete cablata

*TeoriadeiSegnali.it*

## Approccio della esposizione

Gli strumenti usati per aumentare il livello di sicurezza di una rete si basano sulle stesse tecniche usate da un attaccante per scavalcarne le protezioni

La conoscenza dei meccanismi di attacco è fondamentale per poterli prevenire

Gran parte delle protezioni relative all'accesso radio sono le medesime adottabili per la rete cablata

Una architettura di sicurezza è debole quanto il suo componente più debole, pertanto occorre conoscerli tutti

*TeoriadeiSegnali.it*

## Qualità delle password

Le password degli utenti Linux sono salvate nel file `/etc/shadow` nel formato crittografico DES, ed i loro nomi in `/etc/passwd`

Il programma **john the ripper** effettua un *attacco a forza bruta*, confrontando le password crittografate con quelle generate sul momento, a partire da un elenco di possibilità, oppure generando tutte le combinazioni possibili – ma il tempo richiesto **può essere enorme**

Le liste di possibili password possono essere acquistate, oppure **scaricate**

Creiamo un nuovo file con i nomi e le password

```
unshadow /etc/passwd /etc/shadow > mypasswd
```

Eseguiamo il crack

```
john --wordlist=lower.lst mypassword
```

TeoriadeiSegnali.it

## Recupero delle password

Il tentativo di scoprire una password può evidenziare quali di queste sono a rischio di essere scoperte con la stessa tecnica

Il modo più veloce di recuperare una password dimenticata è di *metterne una nuova*, ma

quella vecchia potrebbe essere di per sé buona  
nessuno vorrebbe avere troppe password da ricordare

Esistono

siti on line per il *password crack* e l'MD5 inverso

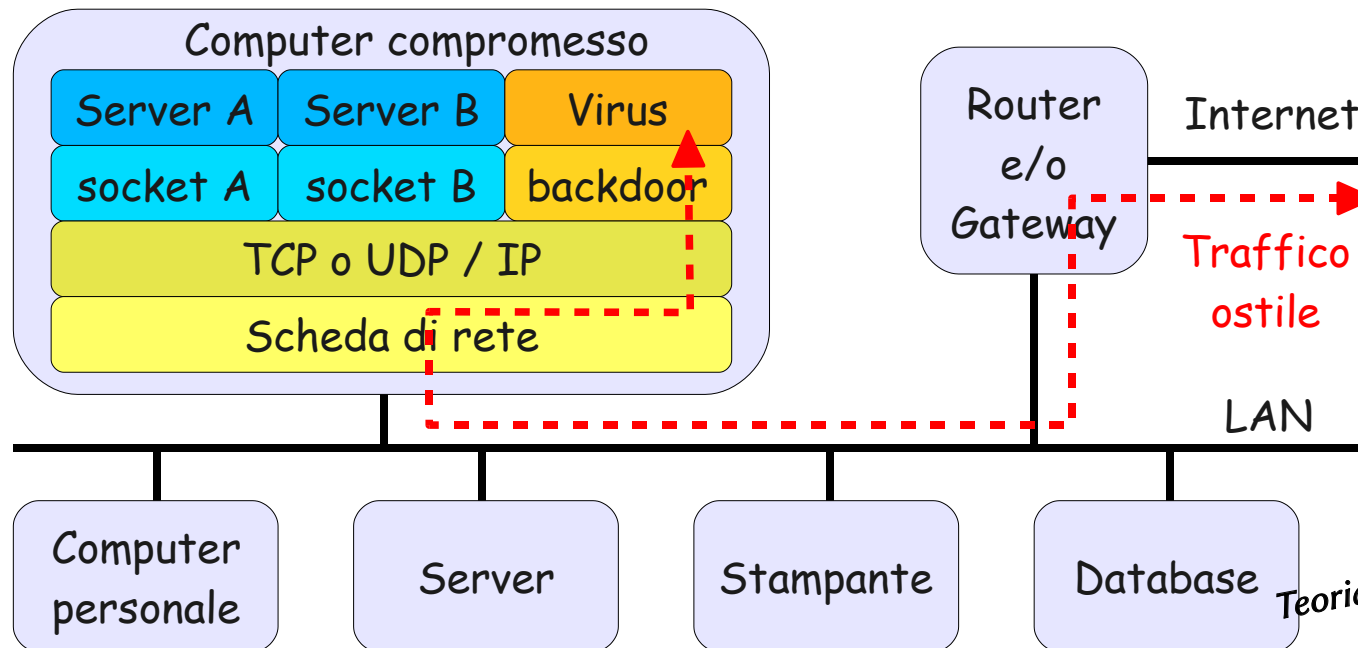
Altri programmi cracker, anche per Windows

Strumenti specifici per il wireless come Aircrack

TeoriadeiSegnali.it

# Effetto dell'intrusione

L'uso delle risorse da parte di un intruso può avvenire scavalcando i meccanismi di autenticazione accedendo così in modo diretto *oppure* facendo eseguire del *codice virale* installato sfruttando la vulnerabilità di un programma



TeoriadeiSegnali.it

## Detezione dell'intrusione

L'uso di programmi e tecniche diagnostiche permette di evidenziare le tracce lasciate dall'intruso, come

anomalie nei file di log → logwatch

files inattesi in /tmp dove tutti possono scrivere

esecuzione di processi sconosciuti, carico di lavoro ed uso della rete anomali → ps, top, vnstat, mrtg

socket di rete in ascolto su porte inattese → netstat, nmap

Per *mascherare* la propria presenza, l'intrusore può sostituire delle *versioni modificate* (mediante un **rootkit**) ai programmi originali modificandone le funzioni, come ad es.

il *processo virale* non è elencato da ps

La *backdoor* non è elencata da netstat

TeoriadeiSegnali.it

## Integrità del sistema

Esistono programmi sviluppati per verificare l'integrità dei propri programmi diagnostici:

`chkrootkit` è in grado di rilevare numerosi rootkit, presenza di worm, alterazioni dei programmi e modifiche ai log di sistema.

`rkhunter` è molto simile, può segnalarci variazioni a file importanti come `passwd`, e le vulnerabilità dei nostri software

L'esecuzione dell'analizzatore può essere resa ricorrente mediante un cron job

Se un sistema è sospetto di essere stato compromesso, i programmi diagnostici dovrebbero essere eseguiti a partire da un **CD di recupero**

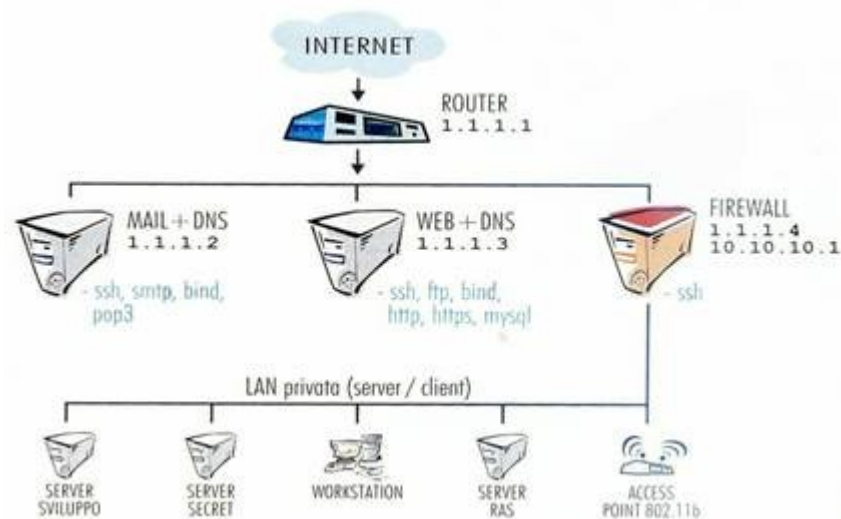
*TeoriadeiSegnali.it*



# Firewall e rete privata

Per **Firewall** in generale si intende un router che interconnette due segmenti di rete e filtra il traffico in modo da far passare solo quello diretto verso gli *indirizzi di trasporto* ammessi

Se il firewall agisce anche come **NAT** (*Network Address Translator*) la rete interna può usare *indirizzi privati* irraggiungibili dal traffico entrante



TeoriadeiSegnali.it

# Firewall di Linux

Il componente del kernel di Linux che gestisce i pacchetti IP che lo raggiungono, permettendo di reinstrararli su di un'altra interfaccia, respingerli, duplicarli, accettarli, è **Netfilter**, che permette di realizzare *un router* mediante un computer Linux

I comandi che permettono di impostare *Netfilter* sono `iptables`, il più completo, ma che ha una sintassi particolare e `ufw` (*uncomplicated firewall*) sviluppato da Ubuntu, più semplice

Le interfacce grafiche a `iptables` e `ufw` consentono di configurare *Netfilter* come **Firewall personale**, bloccando il traffico entrante non diretto verso porte note, ed evitando che del *malware* si ponga in ascolto su di un socket libero

TeoriadeiSegnali.it

# Programmazione del firewall

## Uso personale

Firestarter - permette anche il monitoraggio del traffico e la condivisione della connessione

Gufw – interfaccia grafica di ufw, che a sua volta sostituisce iptables

Kmyfirewall - sviluppato per KDE, coniuga semplicità ed efficienza

Guarddog - raggruppa i servizi da abilitare per classe

## Configurazione di router

Fwbuilder - consente un controllo molto dettagliato di piattaforme firewall anche non Linux, e gira anche su Windows e Mac

Shorewall – non ha interfaccia grafica (se non via webmin), e si basa su files di configurazione

*TeoriadeiSegnali.it*

## Controllo dei socket in ascolto

un comando per verificare, sul proprio computer, quali processi siano in ascolto su quali socket è

```
netstat -n --udp --tcp -p -l
```

elencandone il nome, il PID, la porta, il trasporto, lo stato, se vi sono connessioni attive, e gli indirizzi locale e remoto delle stesse

Un comando che analizza gli altri computer (della propria LAN o anche remoti) allo scopo di verificare su quali porte questi accettano connessioni entranti è **nmap**, la cui sintassi

```
nmap [Tipo di scan] [Opzioni] {specificazione del target}
```

può adattarsi per interrogare interi gruppi di macchine, svolgere verifiche di accensione, analisi dei socket TCP o UDP, limitatamente a intervalli e/o ad insiemi predefiniti...

*TeoriadeiSegnali.it*

## Nmap - esempi

Scansione di 1713 porte note del proprio computer

```
nmap 127.0.0.1
```

Scansione delle porte da 1 a 2000 di un computer esterno

```
nmap -p 1-2000 151.100.122.122
```

Scansione aggressiva di un intervallo di indirizzi IP con  
detezione del sistema operativo e dei servizi presenti

```
nmap -T5 -A -F -sV 151.100.122.50-100
```

Scansione della rete locale per scoprire i computer accesi

```
nmap -sP 151.100.122.0/24
```

TeoriadeiSegnali.it

## Nmap – esito e grafica

Al termine della scansione, nmap riferisce lo stato delle porte come

*aperta*: c'è un programma in ascolto

*chiusa*: è sopraggiunta una risposta negativa

*filtrata*: un firewall impedisce di raggiungere la destinazione

Esiste una interfaccia grafica invocata come **zenmap** che permette di

usare *un wizard* per impostare le opzioni di esecuzione, e salvare il risultato in modo da poterle ri-usare in seguito

salvare il risultato della scansione, e confrontarlo successivamente con quello nuovamente generato

aggregare i risultati di differenti scansioni

fornire una mappa della topologia desunta dall'esecuzione di tecniche di *traceroute*

TeoriadeiSegnali.it

# Vulnerability Scanning & Assessment System

Invece di andare alla ricerca delle intrusioni già avvenute, si può ricercare la presenza degli *elementi deboli*, ossia di quei servizi e configurazioni che faciliterebbero l'intrusione

**Nessus** è uno scanner di vulnerabilità con architettura client-server capace di controllare che per i servizi in esecuzione ed ascolto nei computer della propria rete non siano state diramate segnalazioni di vulnerabilità

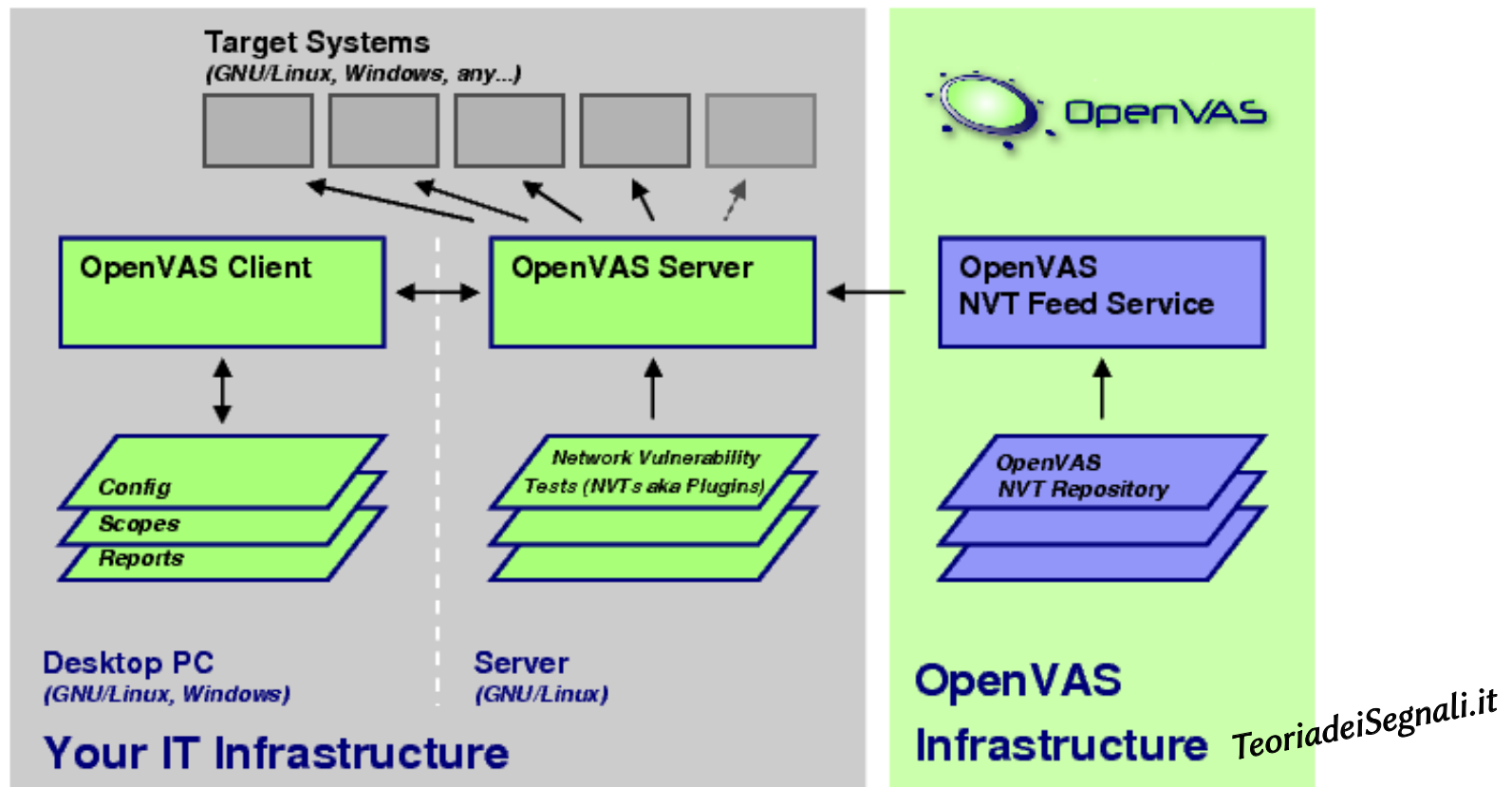
```
sudo service nessusd start  
openvas-client
```

Dopo aver eseguito un *portscan*, Nessus tenta di eseguire gli attacchi elencati in un database di *Network Vulnerability Tests*

TeoriadeiSegnali.it

# Nessus & OpenVAS

Nessus è divenuto proprietario, e **OpenVAS** ne è una fork OpenSource, che mantiene un feed di Network Vulnerability Tests





## Esecuzione di OpenVAS client

Viene configurato un *task* definendo

- Il portscanner da utilizzare
- quali indirizzi e porte verificare
- quali vulnerabilità sperimentare
- credenziali da usare, parametri da adottare

La scansione è eseguita in parallelo sulle diverse macchine, e può essere interrotta selettivamente

Il risultato è memorizzato e confrontabile con altri precedenti

- la sua rappresentazione è orientata alla *comprensione esplorativa* delle vulnerabilità individuate

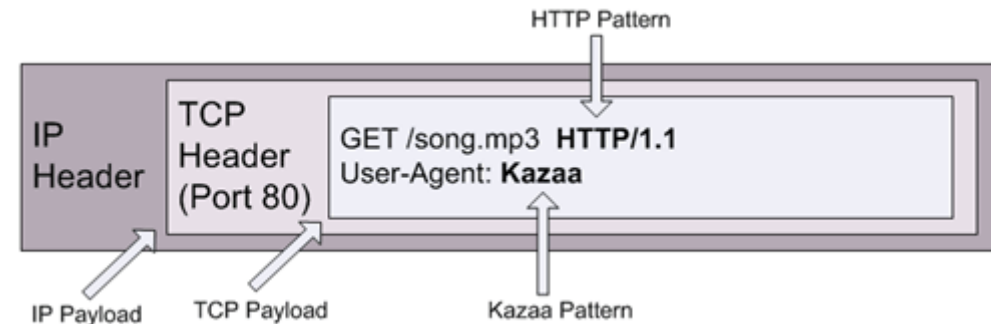
- Può essere esportato in formati navigabili (html e pdf), contenenti i links alle descrizioni più approfondite dei problemi e delle soluzioni

*TeoriadeiSegnali.it*

# Sniffing e Deep Packet Inspection

Anziché eseguire scansioni periodiche, si può monitorare il traffico in *real-time*. Se la *backdoor* utilizza un socket normalmente usato da un servizio legittimo, come ad esempio la porta 80 dell'HTTP, il firewall non può distinguere il traffico *buono* da quello *alieno*

Per accorgersi della intrusione occorre esaminare i pacchetti IP in modo più **approfondito**, oltre l'incapsulamento dalle intestazioni esterne



La libreria **libpcap** permette ai programmi Linux di accedere a *copie* dei pacchetti in transito, complete degli header di qualsiasi livello

L'accesso ai pacchetti visti dalla scheda di rete viene detta **sniffing**, e consente l'analisi manuale del traffico *broadcast* e di quello *diretto* o *uscente* dal computer ospite

TeoriadeiSegnali.it

# Tcpdump e wireshark

La libpcap (*packet capture*) deriva dallo sviluppo di **tcpdump** (sniffer testuale), e viene usata anche da **wireshark** (con interazione grafica)

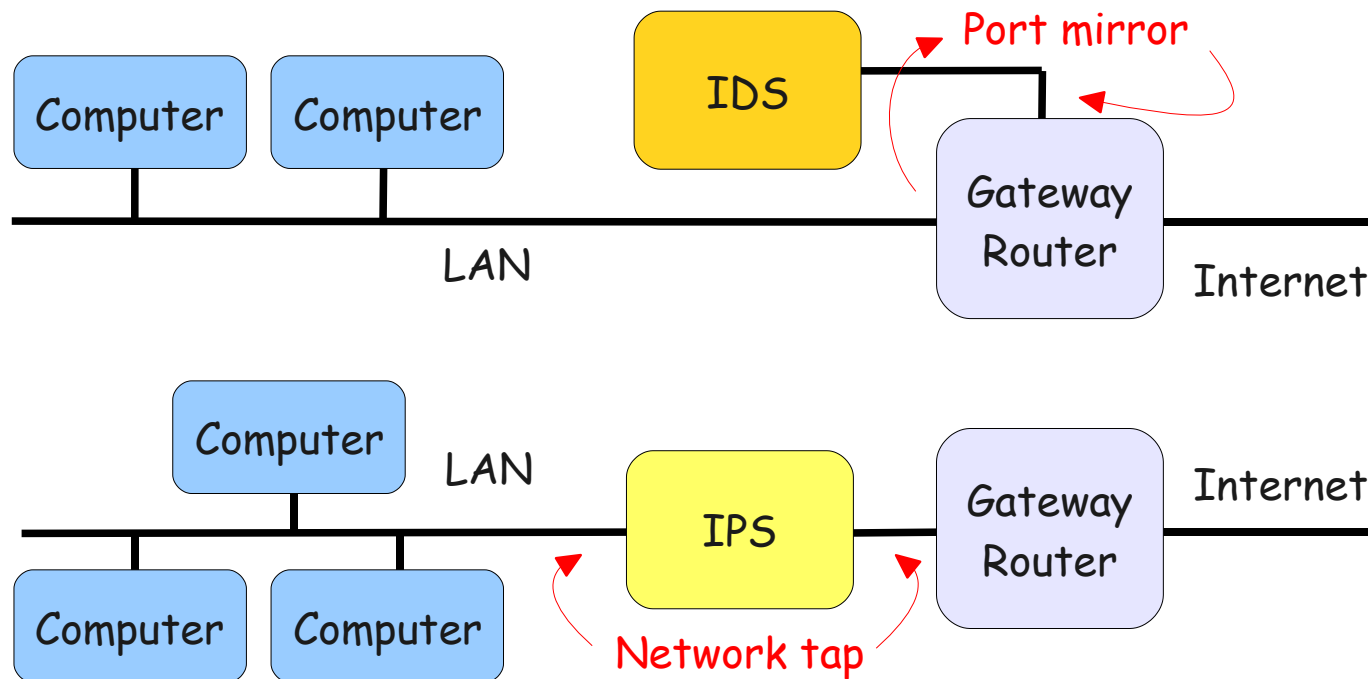
Entrambi permettono di definire dei **filtri** che limitano il traffico catturato in base ad alcuni parametri (ad es gli indirizzi presenti) e di salvarlo in un formato compatibile ad entrambi, per una analisi in differita

Wireshark è provvisto di quasi **100.000 dissettori** che interpretano le intestazioni dei diversi livelli di incapsulamento, permettendo di *letteralmente aprire* i pacchetti ed osservarne il contenuto nei minimi dettagli

Wireshark dispone inoltre di svariati strumenti aggiuntivi orientati ai più comuni protocolli, permettendo analisi statistiche di prestazioni ed efficienza, e facilitando l'analisi visiva delle diverse comunicazioni presenti

# Intrusion Detection & Prevention Systems

Per proteggere non un solo computer, ma l'intera rete locale, occorre che *tutto* il traffico sia analizzato mediante **Port Mirroring** o **Network Tap**



In effetti non è necessariamente così, ma rende bene l'idea

*TeoriadeiSegnali.it*



**Snort** è la tecnologia IDS/IPS più diffusa al mondo

Oltre che come *sniffer*, può essere configurato per realizzare

un **IDS** mediante *port mirroring*, applicando un insieme di regole di ispezione profonda al traffico osservato, e producendo dei files di log che riportano gli eventi triggerati

un **IPS** mediante *network tap*, accedendo al traffico attraverso *iptables*, e controllandolo in modo da inoltrare o scartare in tempo reale i pacchetti, in base ad un diverso insieme di regole

Approfondimenti:

Wikibooks offre una descrizione accurata e ragionevolmente sintetica del suo funzionamento e configurazione come IDS

HTML.it offre una guida per il logging su database, e la visualizzazione degli eventi via interfaccia web

*TeoriadeiSegnali.it*

# AntiSpam e AntiVirus

Lo spam può invitare alla visita di siti da cui si dirama un contagio

I virus sono spesso propagati via email

Una prima difesa contro lo spam è quella di configurare il proprio server SMTP per fargli usare delle **DNS Block Lists** globali

l'indirizzo IP di di provenienza per ogni email entrante è confrontato con quelli di siti segnalati come compromessi

Le email entranti sono quindi analizzate mediante **Spamassassin**

applica un insieme di regole che codificano gli indizi di spam, ed aggiorna le statistiche di un decisore bayesiano

Le email con allegati sono infine analizzate da **Amavisd**

attinge ad un database di virus noti, mettendo *in quarantena* i messaggi a rischio

TeoriadeiSegnali.it

# Sicurezza wireless

Introduzione alle reti 802.11 presso [Wiki di Alef](#)

Definizione del problema presso [Wikipedia](#)

Lista di programmi di analisi e diagnosi presso [insecure.org](#)

Suggerimenti vari presso [1](#), [2](#), [3](#)

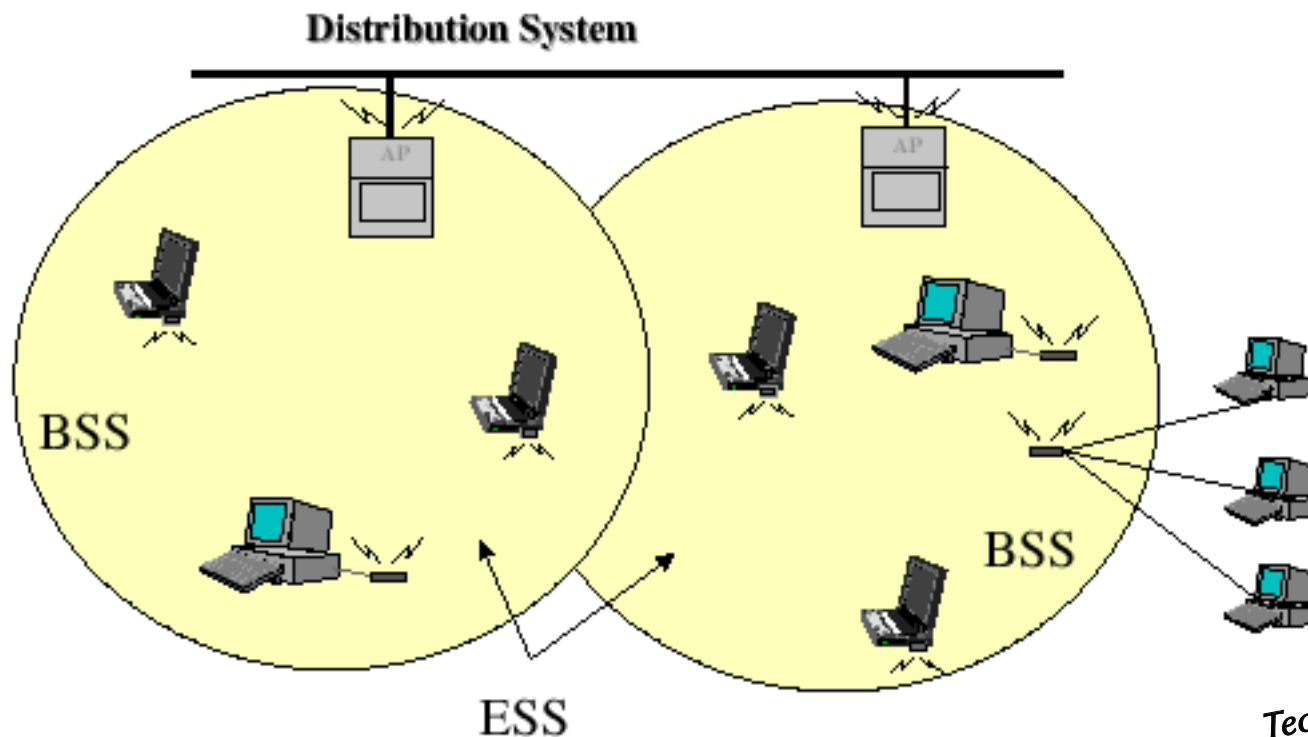
Sniffer wireless: [Wireshark](#) e [Kismet](#) (Linux), [Netstumbler](#) (Windows), [NetSpot](#) (MacIntosh)

Wireless cracker: [Aircrack](#), [Airsnot](#)

*TeoriadeiSegnali.it*

# Estensione radio della rete cablata

Ogni *Access Point* definisce un *Basic Service Set* BSS a cui corrisponde una *IDentità* BSSID, presso il quale un computer si associa

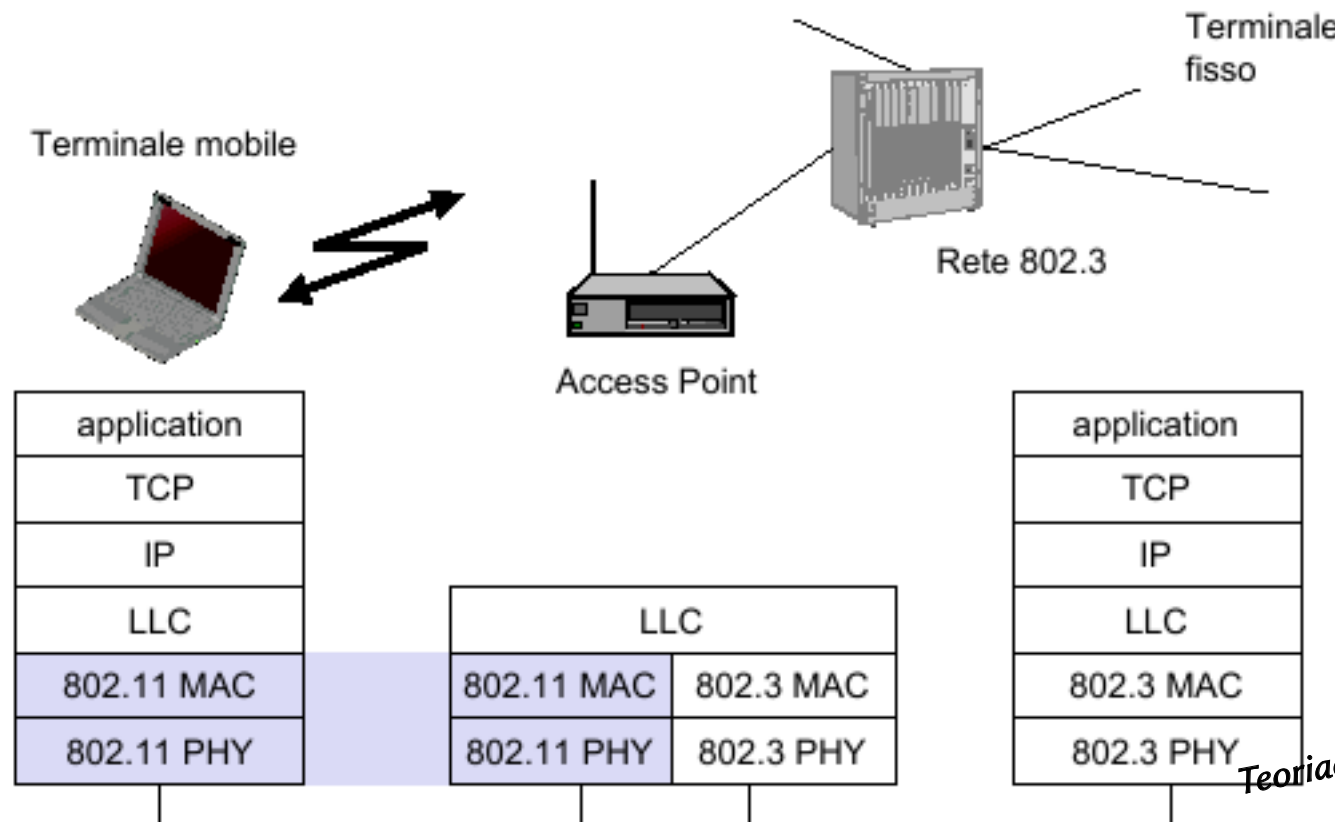


TeoriadeiSegnali.it



# Access Point come tramite

Un *Access Point* svolge un ruolo di *Bridge* tra il *Distribution System* (DS) Ethernet e la diffusione radio Wireless, replicando *in aria* il traffico della LAN



# Wireless Tools per Linux

A partire dal 1996 viene definita e sviluppata **una API** per accedere dallo user space a configurazioni e statistiche delle WLAN, indipendentemente dal suo tipo e da quello del driver, permettendo di modificarne i parametri in modo immediato.

L'API è utilizzata da un insieme di comandi testuali come

iwconfig - manipola i parametri di base

iwlist - permette di individuare AP, frequenze, velocità, chiavi crittografiche...

iwspy - fornisce la qualità del collegamento verso i diversi nodi

iwpriv - per manipolare i parametri specifici dei diversi driver

iwevent - mostra gli eventi prodotti della interfaccia wireless

iwgetid - mostra in forma tersa alcune informazioni sulle interfacce presenti

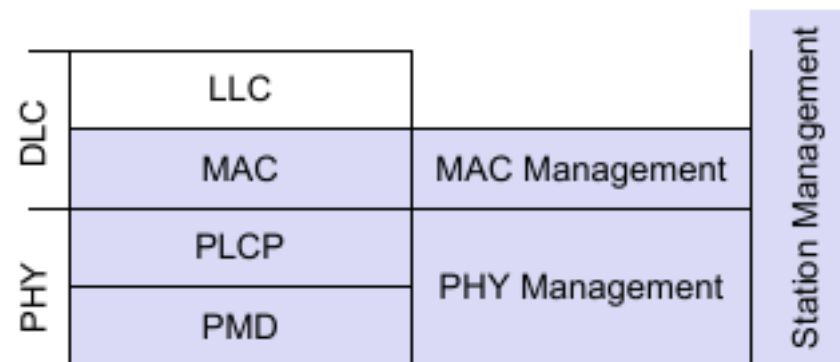
*TeoriadeiSegnali.it*

# Trama 802.11

L'incapsulamento WiFi genera *pacchetti radio* con la struttura



in cui lo strato PLCP (*Physical Layer Convergence Procedure*) offre una interfaccia *omogenea* allo strato MAC (*Media Access Control*) che a sua volta incapsula i dati provenienti dagli strati superiori (*Logical Link Control* LLC, rete, trasporto, applicazione) rispetto alle peculiarità del tipo di trasmissione (a, b, g..) che invece caratterizzano lo strato PMD (*Physical Media Dependent*)



TeoriadeiSegnali.it

# Indirizzi nelle trame 802.11

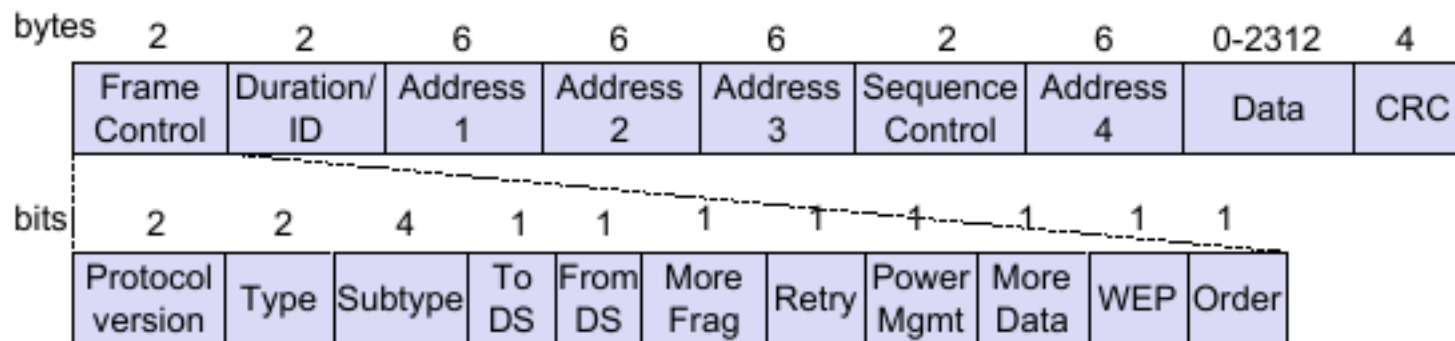
L'intestazione MAC può contenere da due a quattro diversi indirizzi Ethernet, a seconda del contesto del pacchetto:

solo radio: due indirizzi, dell'AP e della stazione mobile

da DS a mobile: tre indirizzi, includendo anche il mittente su rete fissa

da mobile a DS: tre indirizzi, includendo anche il destinatario su rete fissa

da DS a DS: due LAN sono connesse con un ponte radio, e compaiono sia gli indirizzi di rete fissa, che quelli degli AP



TeoriadeiSegnali.it

## Trame 802.11 di gestione e di controllo

Oltre ai *pacchetti dati* della rete cablata *ritrasmessi* per radio, *altri hanno vita limitata tra AP e stazioni associate*, come le *trame di gestione*

**Beacon**, mediante le quali l'AP annuncia il BSSID (il proprio indirizzo Ethernet), il periodo, il canale radio, le velocità supportate, i meccanismi crittografici

**Authentication**, in cui la stazione dimostra di possedere le credenziali e le chiavi crittografiche

**Association**, mediante le quali una stazione richiede e ottiene di vedersi rappresentata presso la rete fissa dall'AP

e le *trame di controllo*

**RTS** e **CTS** che permettono di risolvere il problema del terminale nascosto

**PS-poll**, **Ack**, **CF-end** che permettono di risolvere problematiche energetiche, di integrità, e di contesa di accesso al mezzo

*TeoriadeiSegnali.it*

## Wireless sniffing e modalità monitor

I pacchetti di controllo e di gestione, avendo una semantica tutta interna al contesto radio

- non vengono inoltrati allo strato applicativo

- sono gestiti esclusivamente dal driver dell'interfaccia di rete

- le informazioni contenute nelle loro intestazioni vengono perse

Per poter osservare anche questi pacchetti, prima di eseguire un sniffer come *Wireshark* si deve configurare il driver della interfaccia radio in *modalità monitor*, disabilitando la gestione dello strato di collegamento, e permettendo alle applicazioni di osservare *tutto il traffico radio* che viene ricevuto

In questo caso l'interfaccia di rete Wireless smette di funzionare come tale, e l'eventuale connessione ad Internet tramite l'Access Point è *abbattuta*

TeoriadeiSegnali.it

# Kismet

**Kismet** può funzionare come rivelatore di reti wireless, packet sniffer, e IDS per reti 802.11 a, b e g, operando in *modo passivo* e per questo a sua volta non rilevabile. Rivela:

- la presenza sia di Access Points che di stazioni mobili, individuando le associazioni relative

- la presenza di sniffer wireless attivi (come Netstumbler)

- I BSSID nascosti (per gli AP che non inviano Beacon) in presenza di traffico delle stazioni mobili associate

Salvataggio del traffico con formato compatibile a Wireshark e Aircrack-ng, salvataggio dei BSSID e loro potenze assieme a dati GPS per generare successivamente mappe di ricezione

Architettura *client/server/drone*: il *server* riceve dati provenienti da uno o più *droni*, uno o più *clients* si collegano al *server* per visualizzarli

Interfacciabilità con IDS esterni come Snort

TeoriadeiSegnali.it

## Kismet - 2

Genera allarmi automatici se rivela la presenza di particolari pacchetti o sequenze tipiche per situazioni note

Scansiona i diversi canali 802.11 – ma per osservare tutto il traffico di un determinato AP, si può bloccare su di un solo canale

Produce un risultato testuale

Network List (Autofit)

Name	T	W	Ch	Pkts	Flags	IP Range	Size
! <no ssid>	A	Y	006	75		0.0.0.0	0B
! linksys	A	N	006	50	F	192.168.1.1	0B
! JSMHOME	A	N	001	37		0.0.0.0	0B
! default	A	N	006	72	F	192.168.0.1	0B
. 289554	A	Y	010	5		0.0.0.0	0B

Annotations:

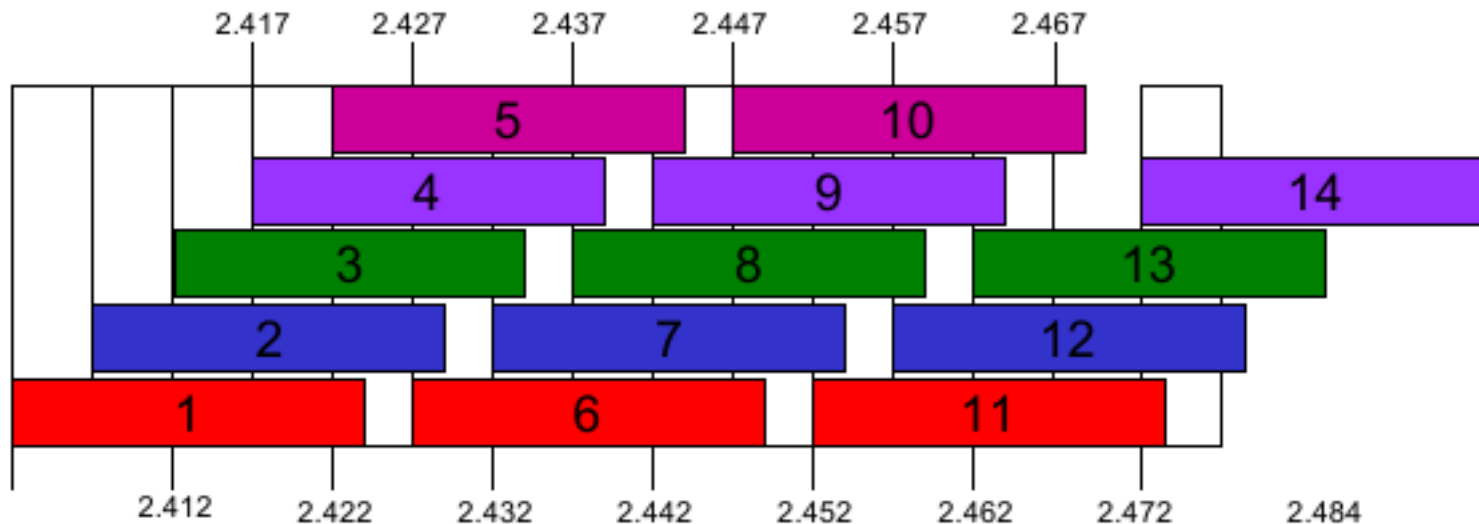
- BSSID (Basic Service Set ID)
- Type of WLAN (A = AP [Access Point], H = Ad hoc, G = Group of wireless networks, D = Data only with no control packets, P = Probe request)
- Is WEP Enabled?
- A# = IP block found via ARP, U# = IP block found via UDP, D = IP block found via DHCP offer, C = Cisco equipment found, F = Vulnerable factory configuration

TeoriadeiSegnali.it



## Canali di trasmissione 802.11b/g

Per l'802.11b/g sono previsti 14 canali a partire da 2412 Mhz, spazati di 5 Mhz; ogni trasmissione occupa 22 MHz, e due AP interferiscono se usano canali distanti meno di 25 MHz



L'assenza di coordinamento porta a violare quasi sempre queste condizioni, ma la modulazione Spread-Spectrum permette il funzionamento contemporaneo anche in presenza di interferenti

*TeoriadeiSegnali.it*

## Kismet - esecuzione

Prima di lanciare Kismet per la prima volta, occorre editare il file `/etc/kismet/kismet.conf` specificando il tipo di interfacce

```
source=ipw2200,eth1,eth1  
source=cisco_wifix,eth2:wifi0,wifi0
```

nel formato `source=sourcetype,interface,name` dove prima si indica il driver, quindi il nome dell'interfaccia, e poi come questa viene individuata viene quindi abilitata l'interfaccia che si intende usare

```
enablesources=wifi0
```

e infine eseguito *drone*, *server* e *client* come utente root

```
sudo kismet
```

se dopo pochi secondi kismet cessa di ricevere pacchetti, occorre *disabilitare il wireless* prima di lanciarlo

TeoriadeiSegnali.it

## Kismet – controllo visualizzazione

Ora che finalmente Kismet è in esecuzione possiamo, premendo il tasto

h – ottenere l'help

s - ordinare gli AP

c – mostrare le stazioni associate all'AP

a – mostra statistiche di potenza ricevuta sui canali

r – grafico del packet/rate ultimi 5 minuti

i – informazioni sull'AP selezionato

*TeoriadeiSegnali.it*

# AirCrack

**Aircrack-ng** è un successore di **AirSnort** (abbandonato) e consiste in uno sniffer e decifratore WEP, WPA e WPA2-PSK, per traffico 802.11 a/b/g

Il tentativo di forzare la chiave di cifratura inizia dopo che sono stati collezionati abbastanza pacchetti

l'uso di metodi avanzati per la ricerca della chiave, permette di ridurre la quantità di dati (e di tempo) necessari – da 5-10 milioni di pacchetti a meno di 100.000

Dopo aver verificato se la propria scheda di rete WiFi è **compatibile**, usiamo il comando **airmon-ng** per

Individuare interfaccia e driver disponibili (**airmon-ng senza opzioni**)

Determinare i processi che possono interferire (**airmon-ng check**)

Porre in modalità monitor l'interfaccia (**airmon-ng start eth1 chan**) specificando il canale chan da utilizzare

TeoriadeiSegnali.it

## AirCrack – Cattura del traffico

La seconda cosa da fare è intraprendere l'**injection test**

```
aireplay-ng -9 eth1
```

che verifica il funzionamento della nostra scheda di rete, inviando sul canale preimpostato dei *broadcast probe request* per sollecitare risposte dagli AP, e quindi valuta la qualità del collegamento

Si possono **ascoltare più canali** in contemporanea:

```
airodump-ng eth1
```

ed individuare se l'AP ha dei client connessi con crittatura WEP

**Salviamo in un file** il traffico prodotto da un AP WEP con client connessi

```
airodump-ng -c 3 --bssid 00:0D:ED:AB:79:65 -w dump eth1
```

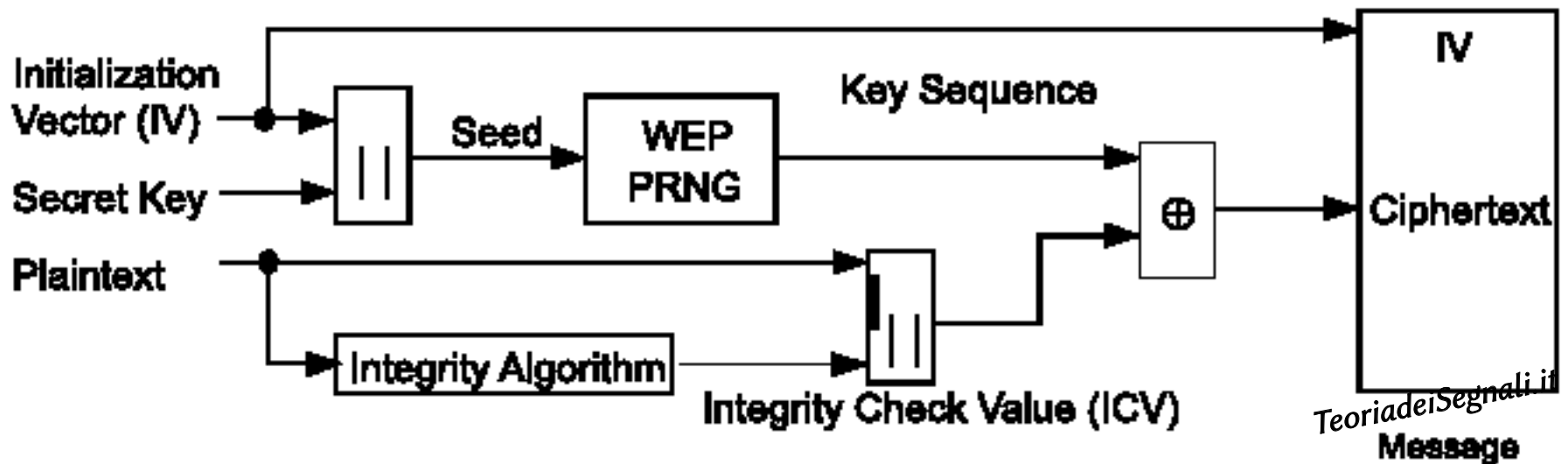
avendo indicato (-c) il canale, l'indirizzo (--bssid) dell'AP ed il file (dump) dove scrivere. Ogni pacchetto contiene un IV

TeoriadeiSegni.it

# Cifratura WEP

Si basa su di una Secret Key di 5 byte, nota ai client, che preceduta da un *Initialization Vector* (IV) di 3 byte costituisce il *seed* di 64 bit posto all'ingresso del *Pseudo Random Number Generator* (PRNG), producendo la *Key Sequence* pseudo-casuale messa in ex-OR con i dati da trasmettere

IV cambia per ogni pacchetto ed è trasmesso in chiaro – aprendo il varco alla possibilità di risalire alla chiave segreta a partire dai dati cifrati



## AirCrack – Scoperta della chiave

una volta collezionati un numero sufficiente di pacchetti eseguiamo

```
aircrack-ng dump-01.cap
```

e dopo un tempo variabile.... ecco svelata la chiave di crittatura WEP!!

Il wiki di Aircrack spiega le tecniche di attacco attivo per accelerare la raccolta dei dati necessari, e come procedere nel caso in cui non vi siano client connessi

Altri tutorial (anche in italiano) mostrano come scoprire chiavi WPA pre-condivise, ossia per le quali airodump-ng osservi una autenticazione di tipo PSK

la sola fase utile per l'attacco è l'*handshaking* iniziale tra Client e AP o aspettiamo che se ne autentichi uno, oppure ne forziamo uno presente a dissociarsi e ri-associarsi

la chiave condivisa così ottenuta viene sottoposta ad un attacco a forza bruta – se fatta bene, è un processo è mooolto lungo!