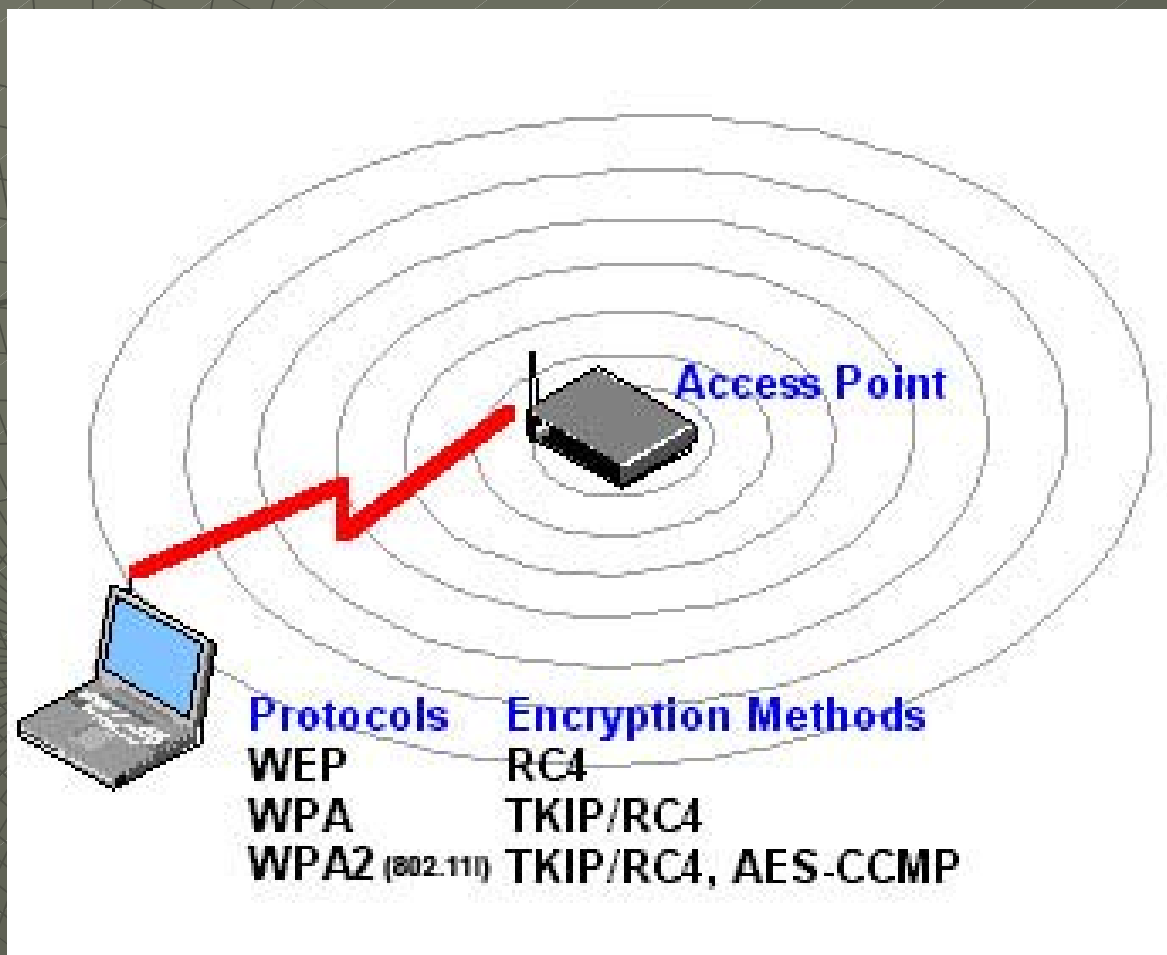


Implementazione di TRIP

Wireless Security e 802.11i

Wireless Security



Wireless network monitoring

- ◆ Kismet
 - Wireless network detector
 - Sniffer (rfmon mode)
 - Intrusion detection system
- ◆ Ethereal
 - Analisi del traffico offline (pcap file)

Kismet e Ethereal

No. ▾	Time	Source	Destination	Protocol	Info
3	0.044379	Motorola_66:d2:f9	Broadcast	IEEE 802.11	Beacon frame, ...
▸ Frame 3 (60 bytes on wire, 60 bytes captured)					
▸ IEEE 802.11					
▾ IEEE 802.11 wireless LAN management frame					
▸ Fixed parameters (12 bytes)					
▾ Tagged parameters (24 bytes)					
Tag Number: 0 (SSID parameter set)					
tag length: 7					
tag interpretation: melissa					
Tag Number: 1 (Supported Rates)					
Tag length: 4					
Tag interpretation: Supported rates: 1.0(B) 2.0(B) 5.5 11.0 [Mbit/sec]					
Tag Number: 3 (DS Parameter set)					

Kismet e Ethereal (2)

Filter: wlan.fc.type == 2 and wlan.fc.wep == 0

No. -	Time	Source	Destination	Protocol	Info
11021	297.18216	192.168.2.51	192.168.2.255	NBNS	Registration NB STUDENT1<OO>
11026	297.35113	192.168.2.51	224.0.0.22	IGMP	v3 Membership Report

▶ Frame 11021 (128 bytes on wire, 128 bytes captured)

- ▶ IEEE 802.11
- ▶ Logical-Link Control
 - DSAP: SNAP (0xaa)
 - IG bit: individual
 - SSAP: SNAP (0xaa)
 - CR bit: Command
 - ▶ Control field: U, func=UI (0x03)
 - Organization Code: Encapsulated Ethernet (0x000000)
 - Type: IP (0x0800)
 - ▶ Internet Protocol, Src Addr: 192.168.2.51 (192.168.2.51), Dst Addr: 192.168.2.255 (192.168.2.255)
 - ▶ User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
 - ▶ NetBIOS Name Service

Kismet e Ethereal (3)

Ether: wlan.fc.type == 2 and wlan.fc.wep == 1

No.	Time	Source	Destination	Protocol	Info
7718	203.59438	192.168.2.51	Broadcast	IEEE 802.11	Data
7745	204.50924	192.168.2.51	Broadcast	IEEE 802.11	Data
7791	205.51040	192.168.2.51	Broadcast	IEEE 802.11	Data

▶ Frame 7718 (68 bytes on wire, 68 bytes captured)

- ▼ IEEE 802.11
 - Type/Subtype: Data (32)
 - ▶ Frame control: 0x4108 (Normal)
 - Duration: 258
 - BSS Id: 00:80:c8:24:33:75 (D-Link_24:33:75)
 - Source address: 00:0f:3d:48:22:29 (192.168.2.51)
 - Destination address: ff:ff:ff:ff:ff:ff (Broadcast)
 - Fragment number: 0
 - Sequence number: 22
 - ▼ WEP parameters
 - Initialization vector: 0x96b100
 - Key: 0
 - WEP ICV: 0xe90a5321 (not verified)
 - Data (36 bytes)

Tipologie di attacchi wireless

- ◆ Probing & network discovery
- ◆ Surveillance
- ◆ Attacchi DOS
- ◆ Impersonation
- ◆ Man in the middle con Rouge AP

Probing & Network Discovery

◆ Active Probing

- Si inviano dei probe con richieste senza SSID per ottenere risposte dagli AP nel range di segnale
- NetStumbler (<http://www.netstumbler.com>)

◆ Passive Probing

- Si ascolta su tutti i possibili canali per i pacchetti TX/RX senza inviare pacchetti
- Kismet (<http://www.kismetwireless.net>)

Surveillance

- ◆ Una volta identificata l'access point vittima si procede con un'analisi specifica del traffico
 - Kismet
 - airodump: 802.11 packet capture program
 - aireplay: 802.11 packet injection program
 - aircrack: static WEP and WPA-PSK key cracker
 - airdecap: decrypts WEP/WPA capture files
(<http://www.cr0.net:8040/code/network/aircrack/>)
 - Chopchop
(<http://www.netstumbler.org/showthread.php?t=12489>)

Attacchi DOS

- ◆ Layer 1: introdurre forti interferenze radio sui canali in cui opera la rete wireless
- ◆ Layer2: packet injection
 - Flooding dei wireless client che sono associati inserendo pacchetti di disassociazione e deautenticazione
 - Void11 (<http://www.wlsec.net/void11>)

Impersonation (MAC spoofing)

- ◆ L'attacker modifica il proprio MAC address con quello acquisito nella fase di *surveillance*
 - Defeating MAC address filtering ACL
 - Il MAC address è visibile anche in una rete WEP encrypted
- ◆ Come si cambia il MAC Address?
 - Linux: `ifconfig eth0 hw ether 01:02:03:04:05:06`
 - Windows: si modifica facilmente nelle proprietà avanzate della propria scheda wireless

Man-in-the-Middle e Rogue AP

- ◆ L'attacker si inserisce nella comunicazione tra client e AP per intercettare, modificare e inoltrare i pacchetti che transitano
 - Rendere inoperativo l'AP legittimo
 - ◆ Attacco DOS layer 1
 - ◆ Attacco DOS layer 2
 - Setting up di un rogue AP che rimpiazza l'AP legittimo
 - ◆ AirJack
(<http://sourceforge.net/projects/airjack/>)

Wireless Attack Detection

- ◆ Access Point Monitoring
- ◆ Wireless Client Monitoring
- ◆ General Wireless Traffic Monitoring
- ◆ Wireless IDS

Access Point Monitoring

- ◆ Informazioni di base: Lista di AP autorizzati (MAC, SSID, canale)
- ◆ Monitoring della rete wireless e salvataggio informazioni in un DB
- ◆ Confronto fra le informazioni raccolte e le informazioni di base
 - Man-in-the-Middle detection: canale non presente in origine, spoofed MAC, ecc

Wireless Client Monitoring

- ◆ Identificare MAC address non validi (non ancora allocati a vendor)
- ◆ Identificare comportamenti anomali: WiFi client che inviano dei probe ma non si autenticano / associano entro un certo periodo di tempo
- ◆ Monitorare il traffico WEP: lo stesso IV non deve essere utilizzato più volte in un intervallo di tempo breve -> sintomatico di un attacco WEPWedgie
- ◆ Tracking del sequence number nell'header 802.11
 - Non deve avere variazioni improvvise -> impersonation attack

General Wireless Traffic Monitoring

- Monitorare eventi come flooding con richieste di tipo:
 - ◆ autenticazione/deautenticazione
 - ◆ associazione/deassociazione
 - ◆ autenticazione sbagliata
- Monitoraggio delle frequenze radio e del rapporto segnale/rumore
 - ◆ Dos attack prevention

Wireless IDS

- ◆ Snort-wireless

<http://www.snortwireless.org>

- ◆ WIDZ

<http://www.loud-fat bloke.co.uk/w80211.html>

- Monitoraggio rouge AP
- 802.11b traffic monitor
 - ◆ Probe
 - ◆ Flooding
 - ◆ MAC – SSID blacklist e whitelist

- ◆ AirIDS

Consigli pratici per configurare una infrastruttura di AP

- ◆ Non annunciare il SSID
- ◆ Non utilizzare modalità Open o Shared senza alcuna protezione
- ◆ Utilizzare l'encryption
 - Non utilizzare WEP
 - Utilizzare WPA e WPA2
- ◆ Se possibile non utilizzare l'encryption con pre-shared key
- ◆ Utilizzare se possibile 802.1x con WPA (enterprise)
 - WPA: 802.1x + TKIP + dynamic key rotation
 - WPA2 (802.11i): 802.1x+AES-CCMP + dynamic key rotation

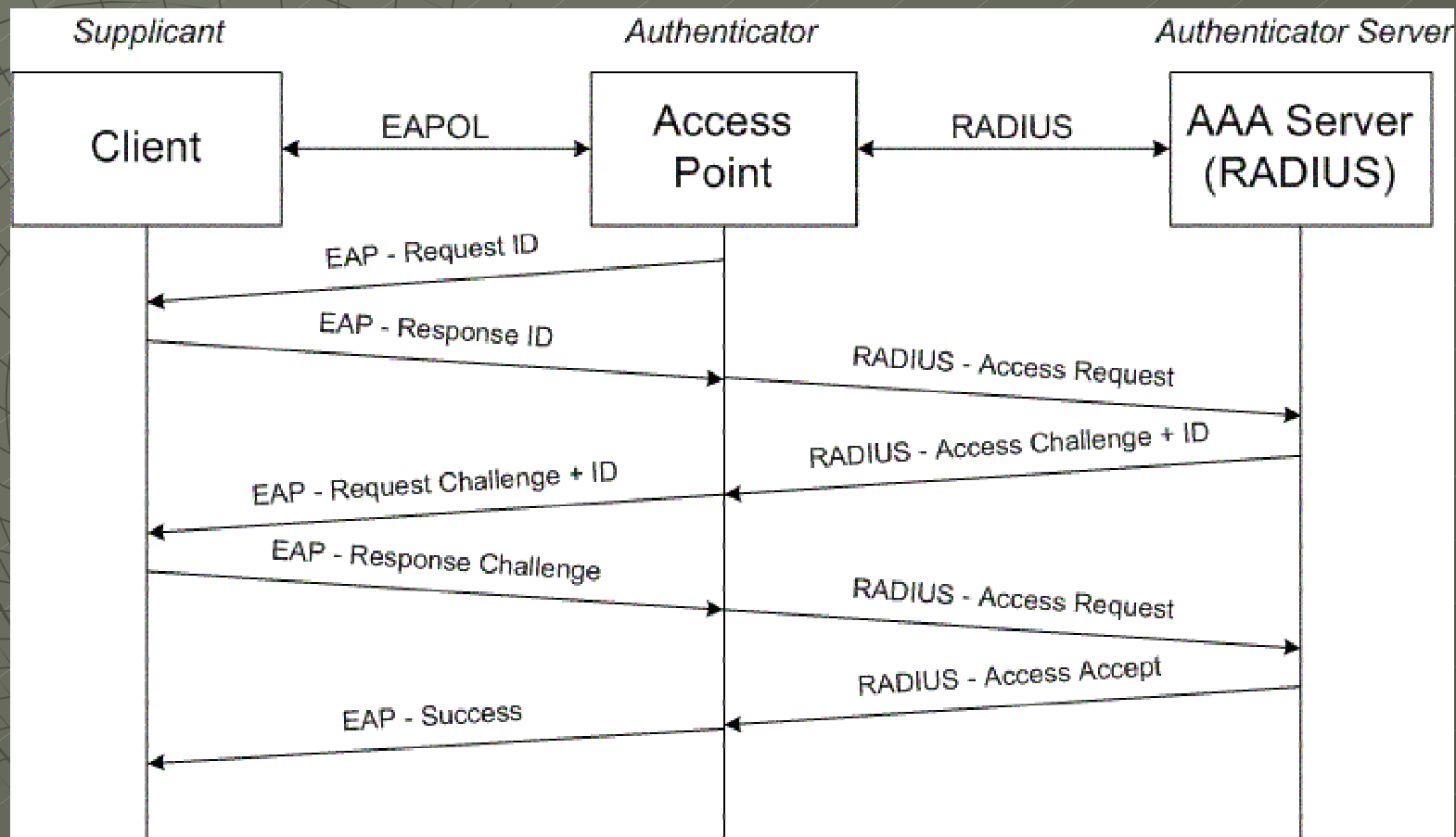
Il progetto TRIP

- ◆ Implementazione di un'architettura hw e sw per un'autenticazione facile dell'utente svincolata dalla struttura ospitante
- ◆ Accesso ai servizi della rete per accedere alla propria sede remota e alle risorse locali essenziali (printing, relay SMTP, ecc)

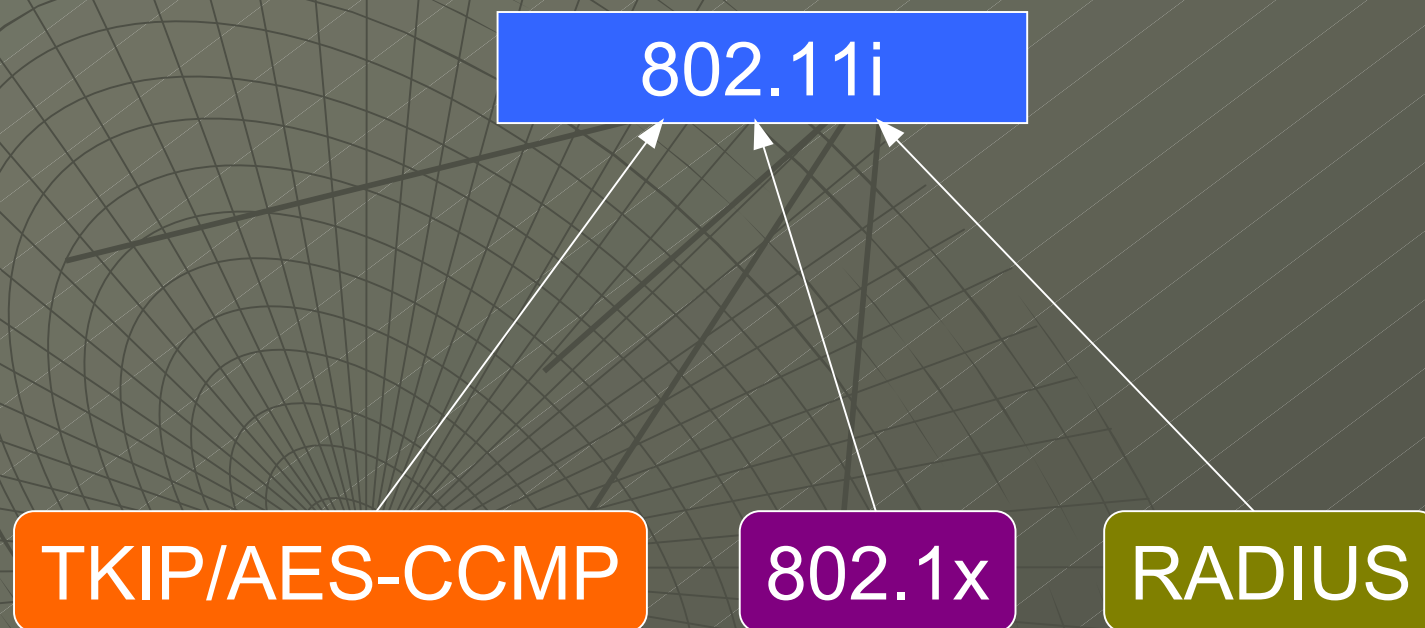


- **802.11i (802.1x)**
- **Portale WEB**
- **VPN**

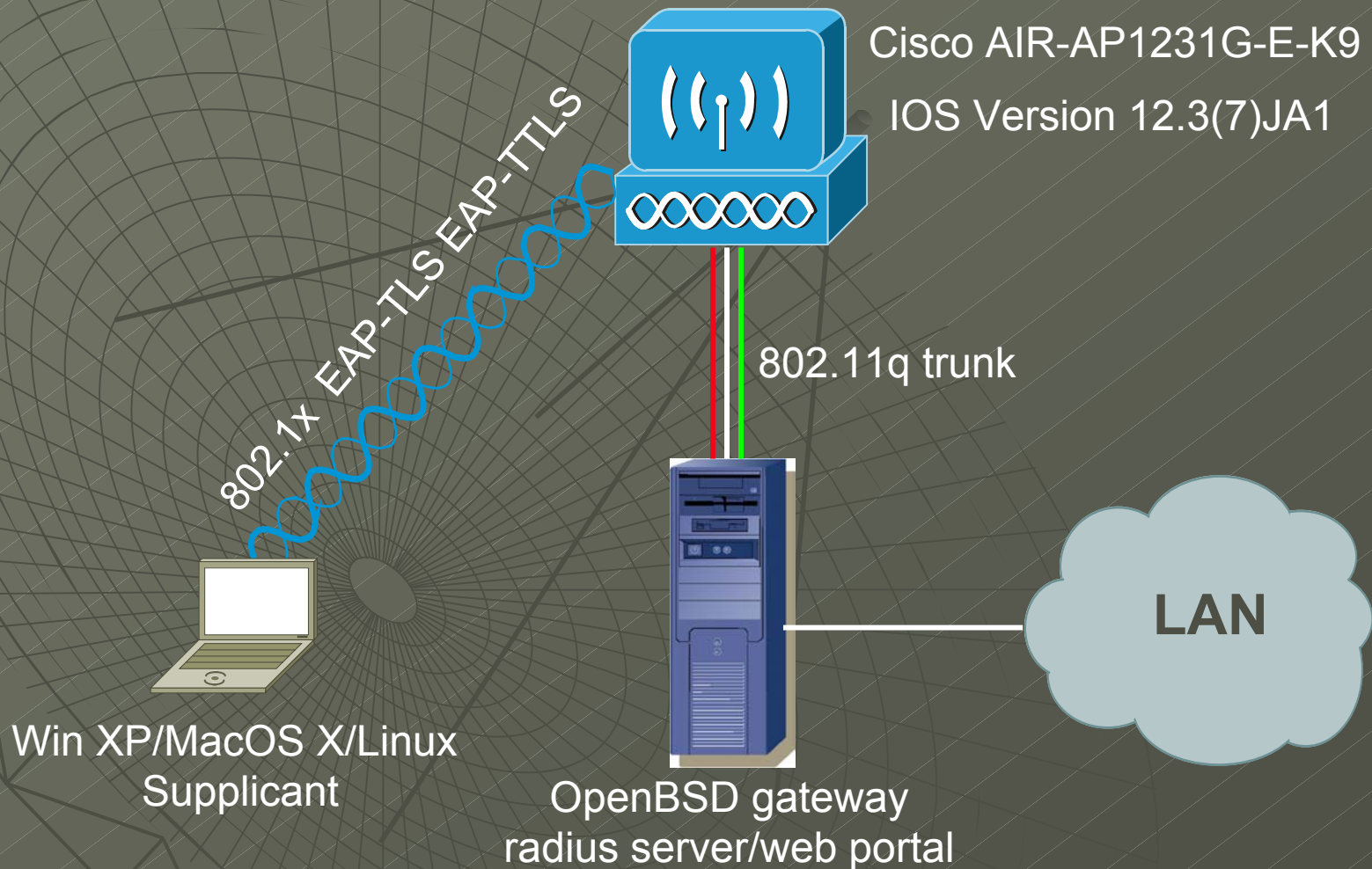
802.1x e EAP



Il protocollo 802.11i



INFN Firenze: architettura TRIP



802.1x: configurazione AP

Administrators

Username	Read-Only	Read-Write
root		✓

Radio0-802.11G SSIDs

SSID	VLAN	Open	Shared	Network EAP
dot1x	1	with EAP		
dot1x-fi	107	with EAP		
web	106	no addtton		
wpa	108	with EAP		

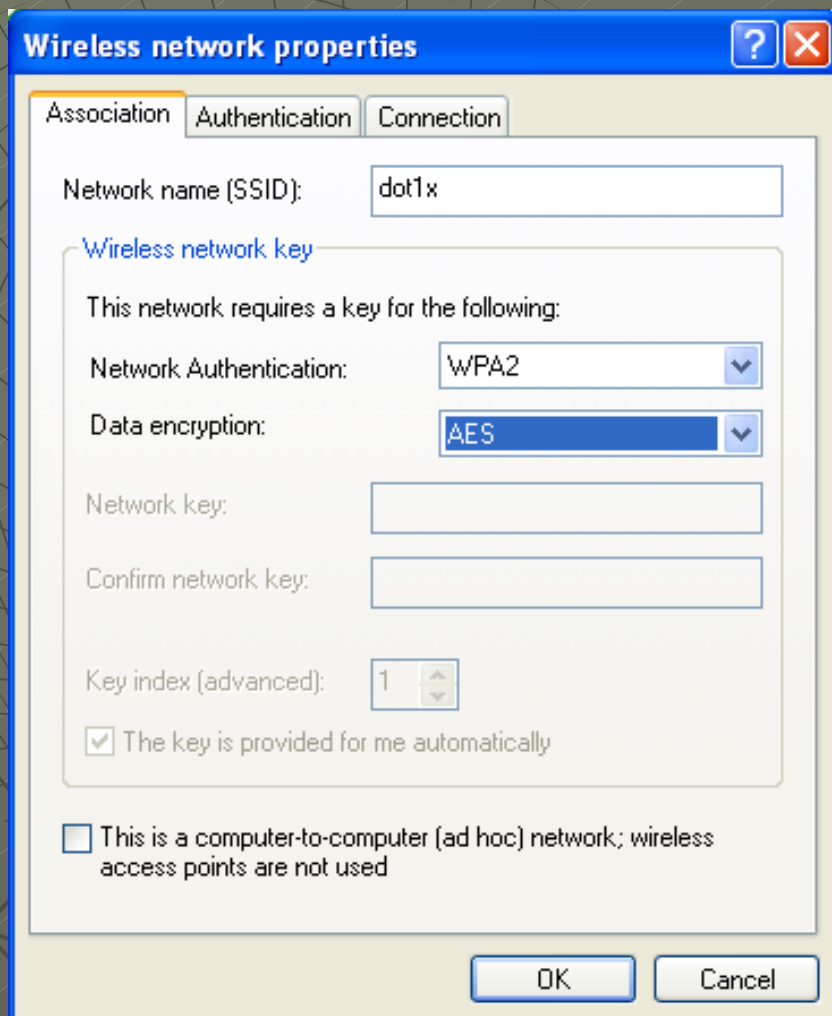
Encryption Settings

VLAN	Encryption Mode	WEP		Cipher						Key Rotation
		MIC	PPK	TKIP	WEP40bit	WEP128bit	CKIP	CMIC	AES CCM	
1	Cipher			✓		✓			✓	✓
106	None									
107	Cipher			✓		✓			✓	✓
108	Cipher			✓					✓	✓

Server-Based Security

Server Name/IP Address	Type	EAP	MAC	Proxy Mobile IP	Admin	Accounting
172.27.7.254	RADIUS	✓				

INFN 802.1x: configurazione XP EAP-TLS

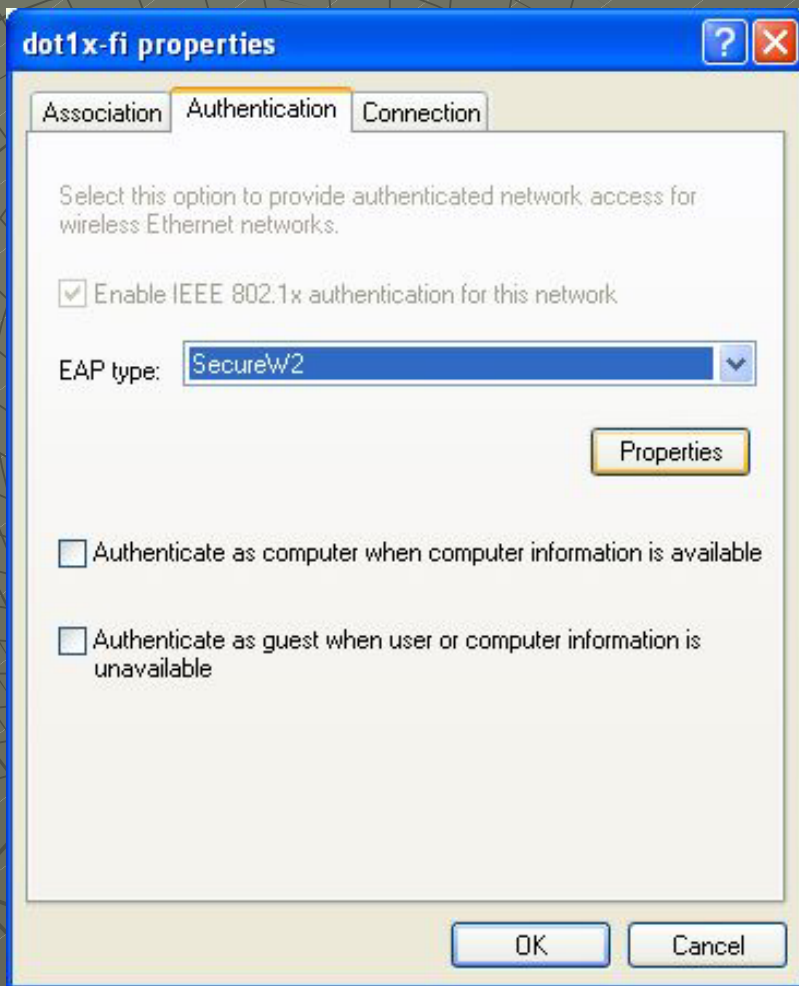


Per il supporto di WPA2 è necessario il seguente upgrade:

<http://support.microsoft.com/?id=893357>

(Francesca Del Corso - INFN FI)

802.1x: configurazione con EAP-TTLS



802.1x: Linux supplicant

- ◆ Atheros driver (Cisco Aironet a/b/g)
 - Supporta WEP e WPA/802.11i
 - Supporto per 802.1x
 - <http://madwifi.otaku42.de/madwifi-cvscurrent.tar.gz>

- ◆ Linux Debian sarge
- ◆ Kernel 2.6.8-2-686
- ◆ Driver Atheros compilato e installato
- ◆ Utilizzo di wpasupplicant

802.1x: Linux e wpa_supplicant

```
#!/etc/wpa_supplicant.conf (version 0.3.8-1)
network={
    ssid="dot1x-fi"
    scan_ssid=1
    key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
    pairwise=CCMP TKIP
    group=CCMP TKIP WEP104 WEP40
    psk="very secret passphrase"
    eap=TTLS PEAP TLS
    identity="*****"
    password="*****"
    ca_cert="/etc/certs/ca.pem"
    client_cert="/etc/certs/perscert.pem"
    private_key="/etc/certs/privkey.pem"
    private_key_passwd="*****"
}
```

OpenBSD gateway

- ◆ È necessario avere 2 NIC una delle quali deve supportare il VLAN tagging
 - fxp0 IP pubblico, interfaccia esterna
 - fxp1 IP privati, 802.1q, interfaccia interna
- ◆ Configurazione VLAN
- ◆ Configurazione freeradius
- ◆ Configurazione dhcpd

Configurazione VLAN

```
fxp1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu  
1500
```

```
address: 00:03:47:b0:9f:e5
```

```
media: Ethernet autoselect (100baseTX full-duplex)
```

```
status: active
```

```
inet 172.27.7.254 netmask 0xfffff00 broadcast 172.27.7.255
```

```
vlan106: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  
mtu 1500
```

```
address: 00:03:47:b0:9f:e5
```

```
vlan: 106 parent interface: fxp1
```

```
inet 172.27.106.254 netmask 0xfffff00 broadcast 172.27.106.255
```

```
vlan107: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  
mtu 1500
```

```
address: 00:03:47:b0:9f:e5
```

```
vlan: 107 parent interface: fxp1
```

```
inet 172.27.107.254 netmask 0xfffff00 broadcast 172.27.107.255
```

Configurazione freeradius

radiusd.conf

- ◆ clients.conf
- ◆ eap.conf
- ◆ proxy.conf
- ◆ users



```
proxy server {
```

```
    synchronous = no
```

```
    retry_delay = 5
```

```
    retry_count = 3
```

```
    dead_time = 120
```

```
    default_fallback = yes
```

```
    post_proxy_authorize = yes
```

```
}
```

```
realm bo.infn.it {
```

```
    type = radius
```

```
    authhost = LOCAL
```

```
    accthost = LOCAL
```

```
}
```

```
realm DEFAULT {
```

```
    type = radius
```

```
    authhost = capetto.fi.infn.it:1812
```

```
    accthost = capetto.fi.infn.it:1813
```

```
    secret = xxxxxx
```

```
    nostrip
```

```
}
```


Configurazione server dhcp

- ◆ Deve essere in esecuzione un'istanza dhcpd per ogni interfaccia fisica o virtuale (VLAN)
- ◆ Bisogna creare una shared network per ogni VLAN

```
/usr/local/sbin/dhcpd vlan106 vlan107 fxp1
```

Portale WEB: TINO

- ◆ Portale web per l'accesso wireless sviluppato al politecnico di Vaasa (fi)
- ◆ Modificato dall'INFN Firenze per aggiungere l'autenticazione tramite certificati digitali
- ◆ Consente l'accesso alla rete wireless previa autenticazione (https) tramite certificato o login/password
- ◆ Utilizza un radius server per l'autenticazione

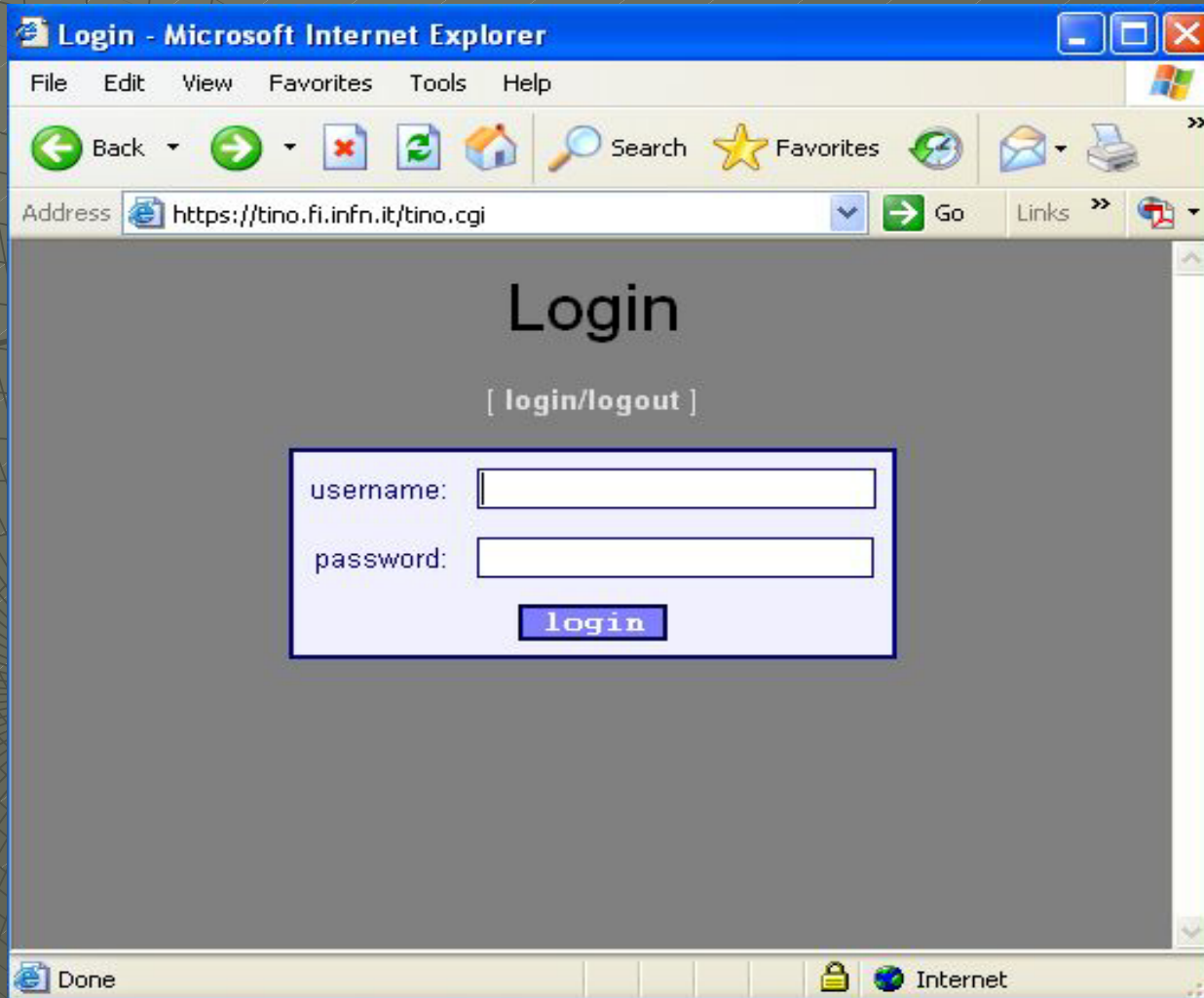
TINO (1)



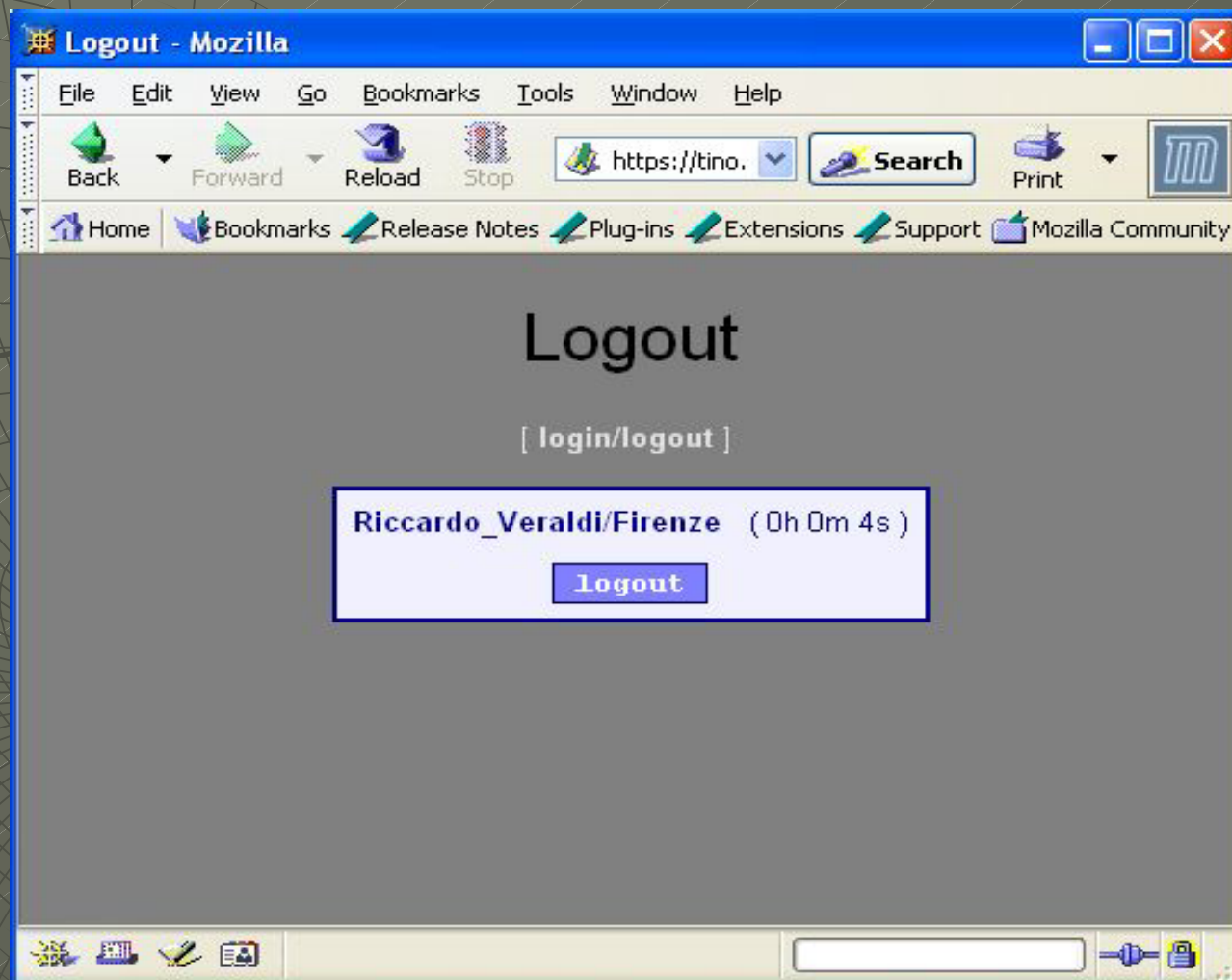
TINO (2)



TINO (3)



TINO (4)



Funzionamento di TINO

- ◆ Shell

- ◆ CGI

- All'utente è presentata una schermata di login
- Il MAC address del computer viene controllato nel file di lease del dhcp
- Se il login ha successo le informazioni rilevanti vengono salvate in un'area di spooling (username,IP,MAC,timestamp)
- TINO chiama firewall.sh per aprire l'accesso ad un determinato IP,MAC,username
- Periodicamente (cron) TINO controlla le informazioni memorizzate nella spool directory e dhcpd.leases
- Se il dhcp lease time è trascorso per un determinato utente/IP/MAC, TINO cancella l'utente dalla spool directory e l'utente deve riautenticarsi

Configurazione di TINO (1)

- ◆ Scaricare la distribuzione con patch modificata per il supporto certificati [tino-INFN-050905.tar.gz](#)
- ◆ Estrarre l'archivio in /usr/local
- ◆ Copiare i file html css in una DocumentRoot del server apache

Configurazione di TINO (2)

Configurazione in tino.pm

```
our $maxtime = 12 * 60 * 60;
our $spooldir = "/var/spool/tino";
our $logfile = "/var/log/tino";
my $fw_script = "sudo /usr/local/tino/firewall.sh";
my $use_syslog = 0;
my $syslog_facility = "local0";
my $syslog_level = "info";
our $ca_match = "/C=IT/O=INFN/CN=INFN Certification
  Authority";
my $dhcpd_leases = "/var/db/dhcpd.leases";
my $radius_host = "127.0.0.1";
my $radius_secret = "1234567890";
our $default_realm = "@fi.infn.it";
our $html_heading = "INFN sez. Firenze";
```

Configurazione di TINO (3)

```
<VirtualHost tino.fi.infn.it:443>
```

```
# General setup for the virtual host
```

```
DocumentRoot /var/www/htdocs
```

```
ServerName tino.fi.infn.it
```

```
ServerAdmin Riccardo.Veraldi@fi.infn.it
```

```
ErrorLog logs/error_log
```

```
TransferLog logs/access_log
```

```
ScriptAlias /tino.cgi "/usr/local/tino/tino.cgi"
```

```
<Location /tino.cgi>
```

```
SSLVerifyClient optional
```

```
SSLOptions +StdEnvVars
```

```
SSLOptions +ExportCertData
```

```
SSLVerifyDepth 2
```

```
</Location>
```

Configurazione TINO (4)

- ◆ Scrivere un file script di firewall personalizzato, secondo la seguente forma:

```
open <IP> <MAC> <USERNAME>
```

```
close <IP> <MAC> <USERNAME>
```


TINO (5)

- ◆ Creare una cron entry per TINO

```
USER=www
* * * * * /usr/local/tino/tinocheck.sh

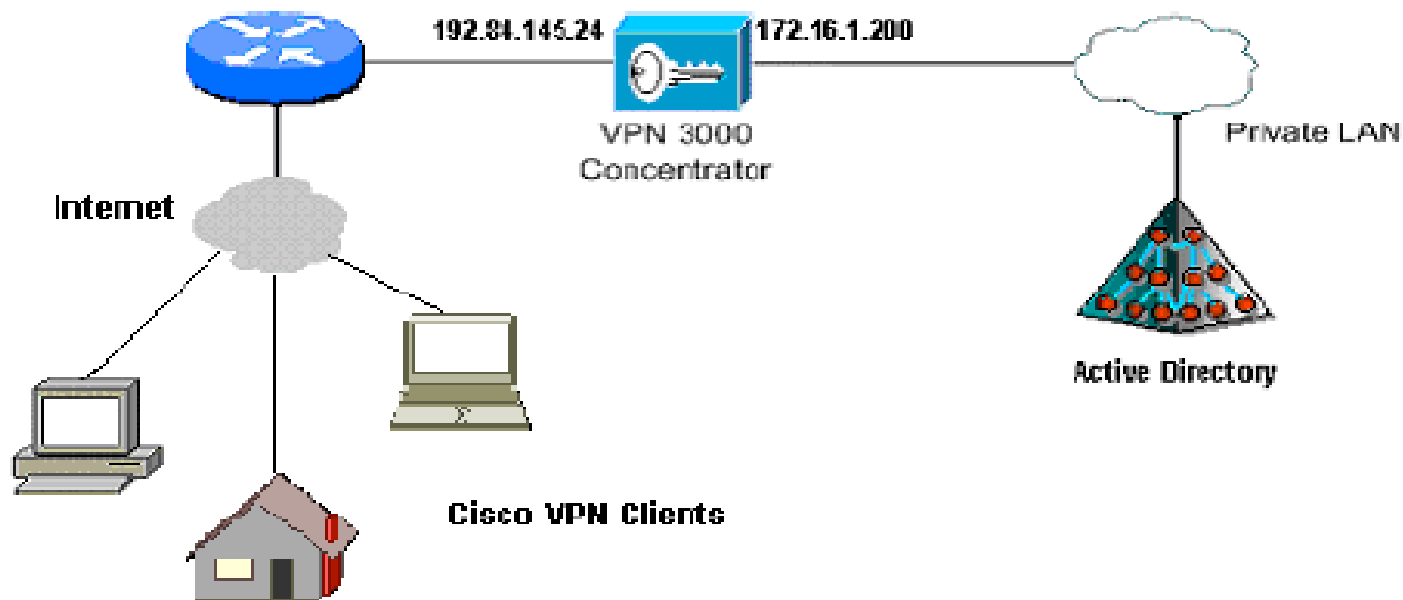
#!/bin/sh
# /usr/local/tino/tinocheck.sh
/usr/local/tino/tino check
```

- ◆ Configurare *sudo* per eseguire lo script di firewall come user www
- ◆ Allo startup eseguire
`/usr/local/tino/tino reset`

VPN: autenticazione con certificati

<http://www.Inf.infn.it/sis/preprint/pdf/INFN-TC-04-17.pdf>

- ◆ **INFN sez. Firenze**
- ◆ **INFN sez. Bologna**
- ◆ **INFN CNAF**



Conclusioni

- ◆ L'utilizzo di 802.1x con WPA e certificati consente una comunicazione wireless sicura
- ◆ Il portale Web https è una soluzione di compromesso accettabile per i sistemi che non supportano 802.1x
- ◆ L'autenticazione RADIUS può essere demandata ad una struttura di server proxy svincolando l'autenticazione dalla struttura che ospita l'utente