

War Driving

- related fixed line attacks
 - war dialing
 - port scanning
- war driving
 - drive through a metropolitan area looking for wireless access points
 - software logs, latitude/longitude
 - runs software near each point to probe vulnerabilities

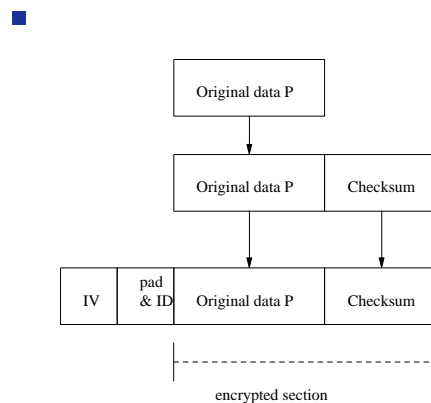
BSS and IBSS

- core of an 802.11 network is called a Basic Service Set (BSS)
 - consists of an access point (AP)
 - clients (stations) connect to the AP
- APs send beacons giving SSID
 - station and AP go through authentication to connect
- station should *disassociate* itself from the AP
 - an AP will timeout if a station just leaves the area

WEP

- Wired Equivalent Privacy (WEP)
- encryption using WEP
 - uses a shared key (K) mechanism with a symmetric cipher called RC4
 - key size is either 40 or 104 bits
- transmitting device
 - creates 24 bit random *initialization vector (IV)*
 - uses IV and K to encrypt the data $E(K,IV,P,C)$
 - transmitted to receiver who then uses shared key, K, and IV to decrypt data

WEP



WEP

- notice that IV is not encrypted
- some vendors claim 40 bit or 104 bit encryption
- some vendors claim 64 bit or 128 bit encryption
 - misleading as 24 bits are sent in plaintext
 - thus 40 bit and 64 bit claims are the same
 - and 104 and 128 bit claims are the same

Authentication using WEP

- upon association the station and AP exchange type of authentication required
 - selecting *open* indicates no authentication
- selecting *shared secret* requires authentication

Authentication using WEP

- authentication can be done using the reverse of encryption
- station, A, sends a *nonce* (random number) to another station, B
 - station B encrypts the datum (with A's key, K) and returns it to A
- if A can decrypt the datum then it notifies station, B, that authentication was successful

Problems with WEP

- WEP is not totally secure
- key management, using symmetric key creates a problem
 - how do we send a machine the key safely?
- a 40 bit key is small and can be broken by brute force on commodity PCs
- furthermore RC4 as implemented in WEP is weak and can be cracked if enough traffic is intercepted see [this paper](http://www.crypt0.com/papers/others/rc4_ksaproc.ps) `<http://www.crypt0.com/papers/others/rc4_ksaproc.ps>`
- also see [airsnort](http://airsnort.shmoo.com) `<http://airsnort.shmoo.com>`

Problems with WEP

- some vendors have implemented the IV choice in the same predictive manner
- some vendors do not choose a different IV !

Snooping/Eavesdropping

- trivial to implement on GNU/Linux
- many programs exist
 - etherfind, snoop
 - tcpdump
 - easy write your own to capture passwords
- passwords are often transferred in plaintext in some protocols
 - ftp, telnet, rlogin, www

URL format

- *protocol / /user:password@host:port/path*
- protocol: http, ftp, news
- user and password: optional, but in plaintext!
- port: normally 80
- host: FQDN for remote servers
- path: relative to the servers root

Socket code to read all packets

- ```

/* OPEN PROMISCUOUS SOCKET */
if ((sd = socket (AF_INET, SOCK_PACKET,
 htons (ETH_P_ALL))) < 0) {
 perror ("Can't get socket: ");
 exit (1);
}

/* SET PROMISC */
strcpy (oldifr.ifr_name, device);
if (ioctl (sd, SIOCGIFFLAGS, &oldifr) < 0) {
 close (sd);
 perror ("Can't get flags: ");
 exit (2);
}

```

## Socket code to read packets

```

■ ifr = oldifr;
 ifr.ifr_flags |= IFF_PROMISC;

 if (ioctl (sd, SIOCSIFFLAGS, &ifr) < 0) {
 close (sd);
 perror ("Can't set flags: ");
 exit (3);
 }

 while (TRUE) {
 /* This is the main data-gathering loop */
 sizeaddr = SN_RCV_BUF_SIZE;

 length = recvfrom (sd, buf,
 SN_RCV_BUF_SIZE,
 0, &saddr, &sizeaddr);
 if (length > 0)
 handle_frame (buf, length, &saddr);
 }

```

## Socket code to read all packets

- under GNU/Linux the user needs to run the above code as root
- provides no security as an attacker could run it from their laptop

## handle\_frame

- function decodes the message held inside buf and looks for plaintext password in ftp, telnet, URLs
- it could search for particular
  - src ip address
  - dest ip address
  - ftp packets
  - technique known as packet filtering
- with advent of very fast inexpensive machines and large RAM sizes, capturing packets is easy

## Security at the data link layer

- 802.11i, is an amendment to the IEEE 802.11 standard specifying security mechanisms for wireless networks
  - created due to the failure of WEP
- ratified on 24 June 2004, and supersedes the previous security specification, Wired Equivalent Privacy

## Security at the data link layer

- 802.11i uses AES-based CCMP to provide confidentiality, integrity and origin authentication
  - AES uses a 128 bit key
- CCMP is a mode of operation for cryptographic block cipher providing authentication and privacy
- makes use of the Advanced Encryption Standard (AES) block cipher, whereas WEP and WPA use the RC4 stream cipher

## Security at the data link layer

- Wi-Fi Protected Access (WPA and WPA2) is a class of systems to secure wireless networks
  - WPA requires either new hardware or existing hardware to be firmware upgraded
    - WPA2 cannot use older WEP hardware
- WPA implements the majority of the IEEE 802.11i standard
- WPA2 implements the full IEEE 802.11i standard

## Legacy hardware

- problem exists, huge amount of WEP hardware deployed
  - might like to upgrade or reflash existing hardware
  - but some hardware cannot be firmware upgraded
- we could use TKIP

## Legacy hardware

- Temporal Key Integrity Protocol or TKIP is a security protocol used in the IEEE 802.11 wireless networks
- recall the problems of WEP
  - too short a key (40 bits)
  - IV (initialisation vector) is predictive, making it susceptible to replay attacks
- the key used for encryption in TKIP is 128 bits

## Legacy hardware

- TKIP also changes the key used for each packet
- it does this by incrementing a unique 48 bit serial number which is used as the IV and also as part of the key
- TKIP also generates part of the key from its base key
  - which in turn is created every time a station associates to an access point
  - all these elements combined will stop a replay attack

## Conclusion

- WEP is broken, do not trust it
  - WEP legacy hardware will live for many years to come
- we should move to WPA2
  - practically we should use TKIP and WPA in the interim
- also we should try and use application and transport security wherever possible ( ssh and openssl for example)