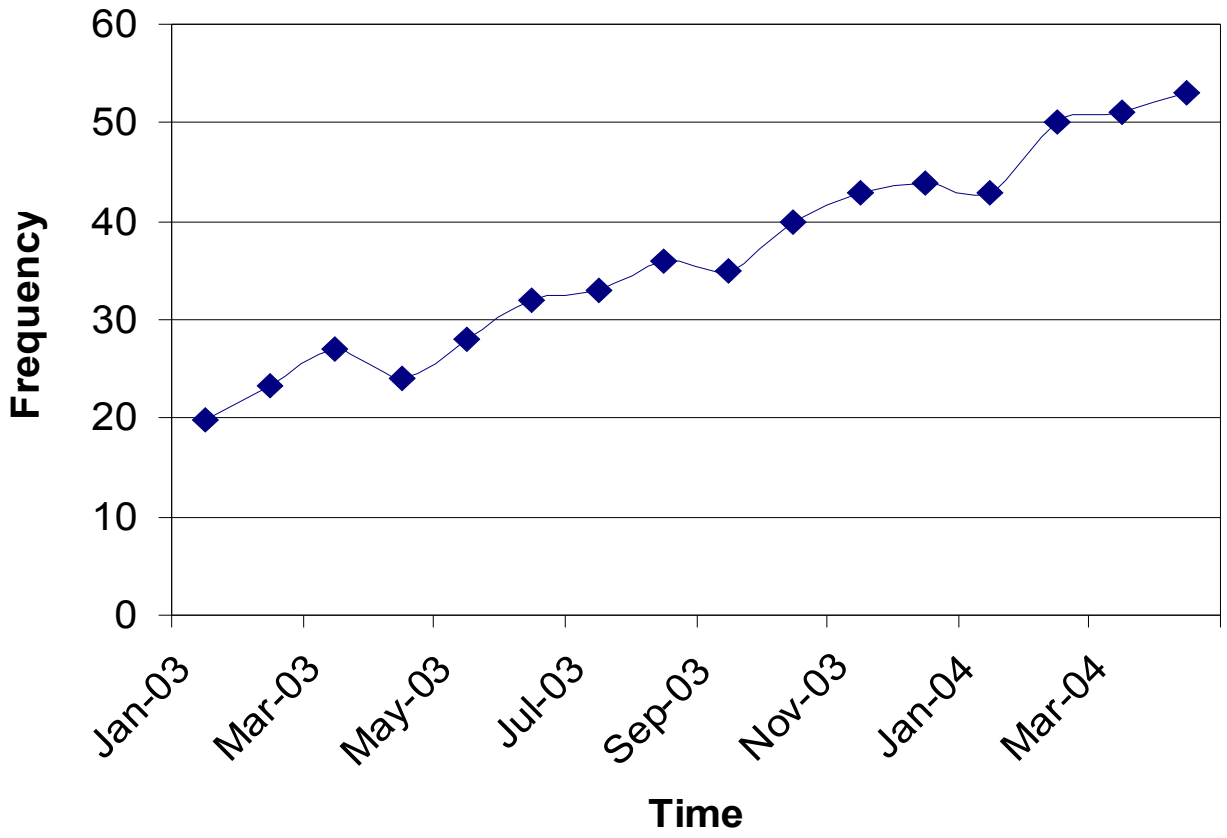
Detecting Detectors: Layer 2 Wireless Intrusion Analysis

Joshua Wright

jwright@sans.org

Obligatory Graph



WiFi Proliferation

- N. Miami Beach PD replaces CDPD network with Metro-WiFi
- Truckers get Internet access through Truckstop.net and Sprint
- WiFi on the Autobahn in Agip Gas Stations
- St. Paul Winter Palace ice castle offers WiFi to "anyone nutty enough to lug along a laptop" (1/2004 carnival: -20 farenheit)

Today's Goals

- Introduce the topic of wireless LAN intrusion analysis
- Explain the 802.11 MAC specification
- Identify common WLAN attack methods
  - Tools, packet captures and detects
- Discuss how we can identify these attack tools on our networks

802.11 Overview

802.11 Specification

- Multiple components
  - Physical transmission and modulation
  - MAC-layer specification for framing
  - Quality of service components
  - CIA of data (security)
- WLAN IDS analyst is mostly interested in MAC-layer data

MAC Layer Provisions

- Basic access mechanisms (CSMA/CD)
- Fragmentation support
- Reliable data delivery
- Network separation on the same frequencies (BSSID)
- Mobility between BSSs
- "Privacy"
- Power Management

Terms

- RFMON – Radio Frequency Monitoring
  - Describes the collection of raw 802.11 frames
- BSS – Basic Service Set
  - A logical grouping of stations on a wireless network with an AP
- IBSS – Independent BSS
  - An "ad-hoc" network, peer-to-peer communication
- ESS – Extended Service Set
  - Multiple BSS's that support roaming

More Terms …

- SSID – Service Set Identifier
  - The "network name" chosen to group AP's
- Open vs closed
  - Referring to the authentication of a client on the WLAN
- Authentication and association
  - Stages of connecting to the WLAN
- STA
  - Short for a station

WEP In Brief

- First IEEE 802 mechanism to introduce authentication at the network level
- Prompted development into 802.1X for generic network authentication
- Was not meant to protect from unauthorized access

Breaking WEP

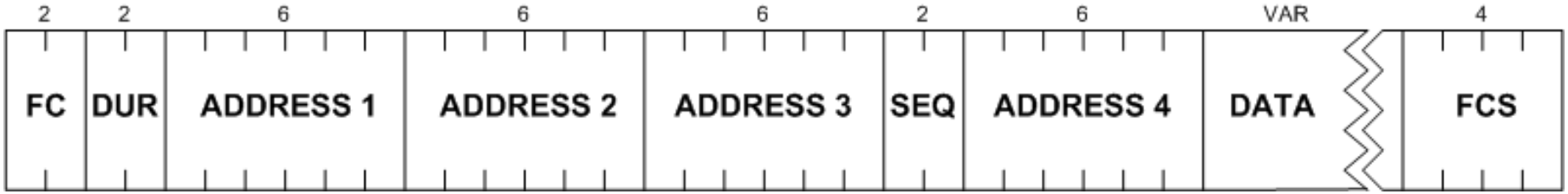
- WEP uses RC4 stream-cipher
- Originally used a 40 bit key (104, 232)
- Shared key XOR'd with plain text to generate cipher text
- Uses an initialization vector (IV) in cleartext
- First byte is constant, can derive WEP keys from millions of packets (B+3):ff:N

Cracking WEP with AirSnort

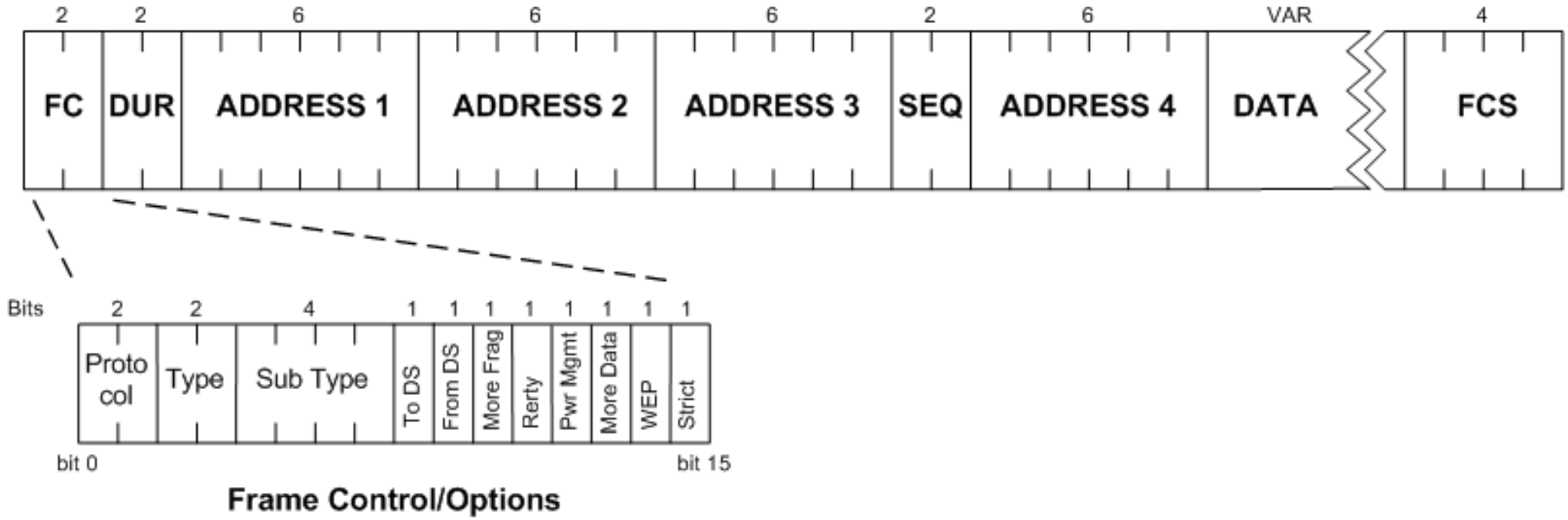
- AirSnort is a Linux tool that utilizes the FMS weaknesses to recover WEP keys
- Requires millions of packets (hours to days of traffic collection)
- Collects traffic in RFMON mode – cannot be detected at layer 2
- Once a key is compromised, all bets are off (even with 802.1x)

802.11 MAC Framing

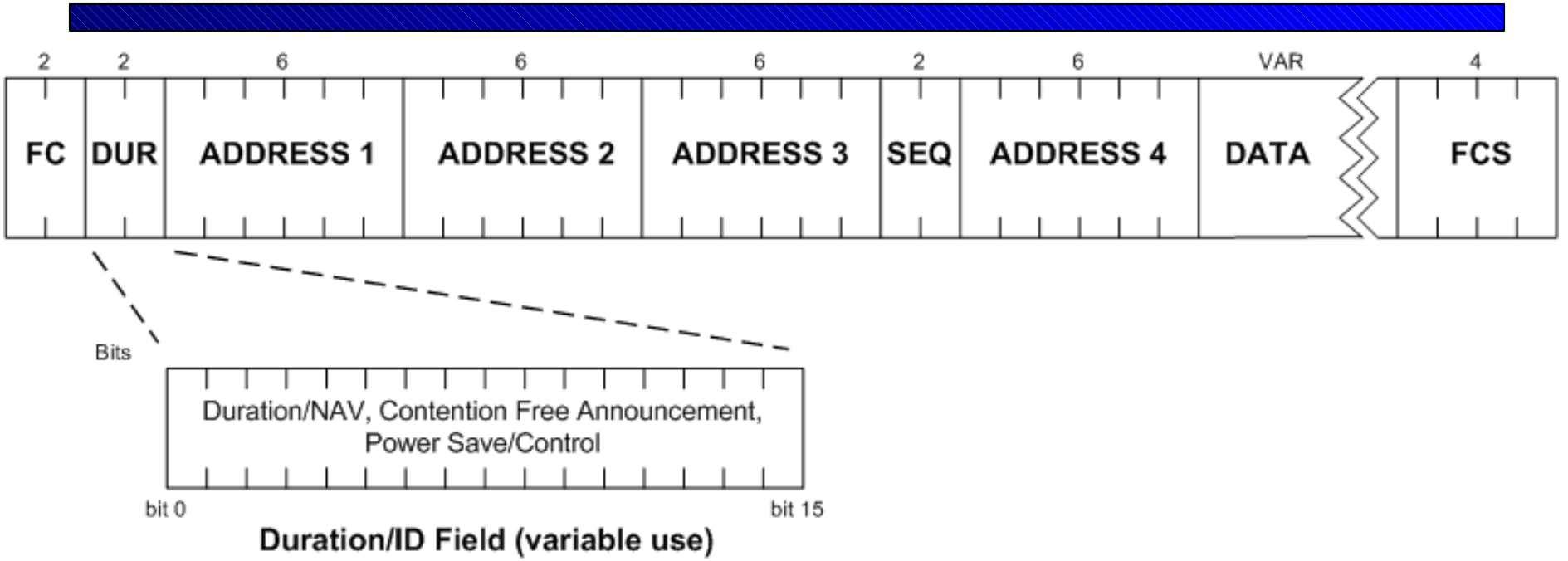
Generic 802.11 frame header



802.11 Frame Control Field

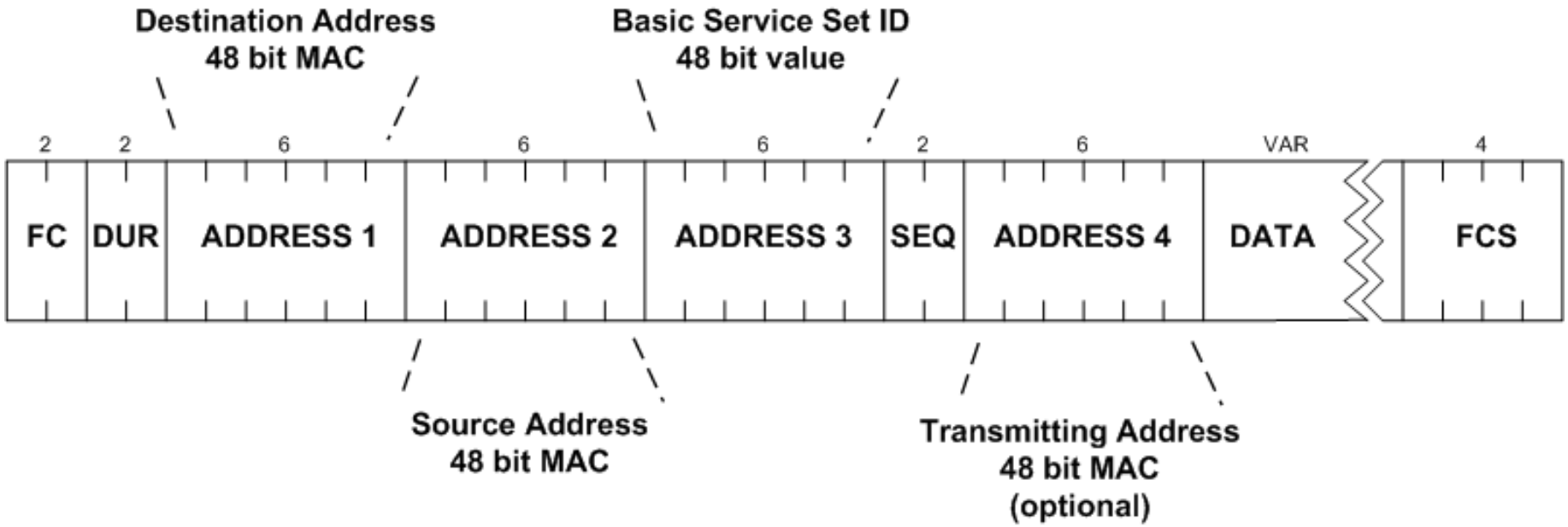


802.11 Duration/ID Field

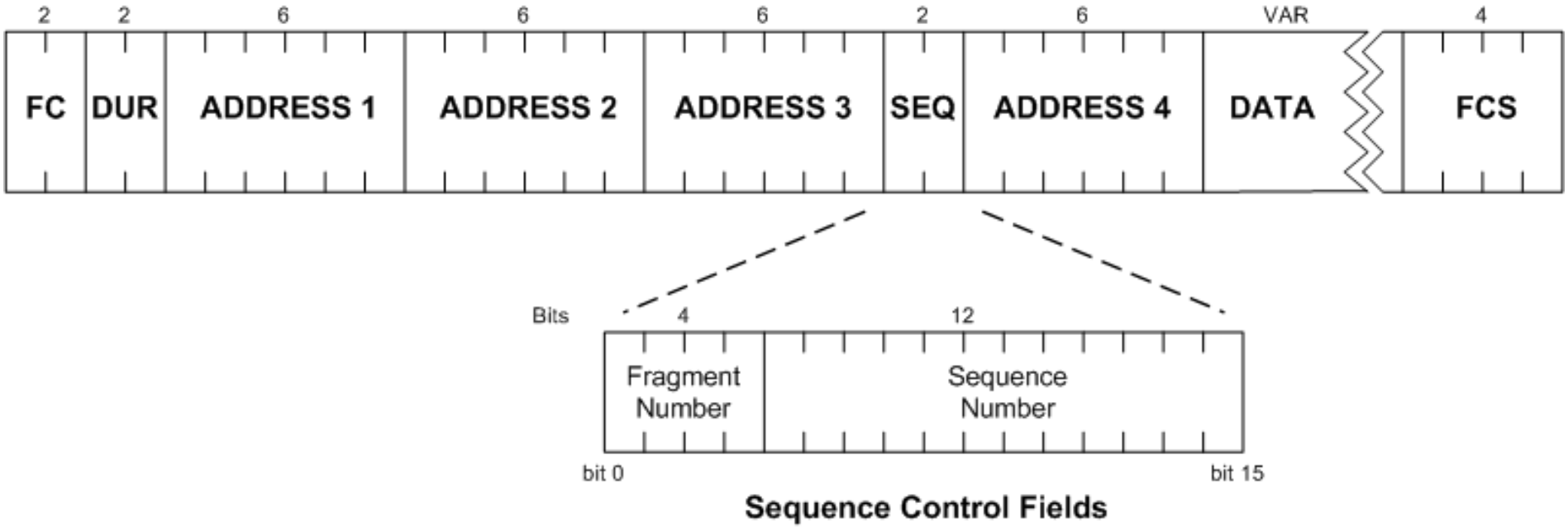


Duration/ID Field (variable use)

802.11 Addressing

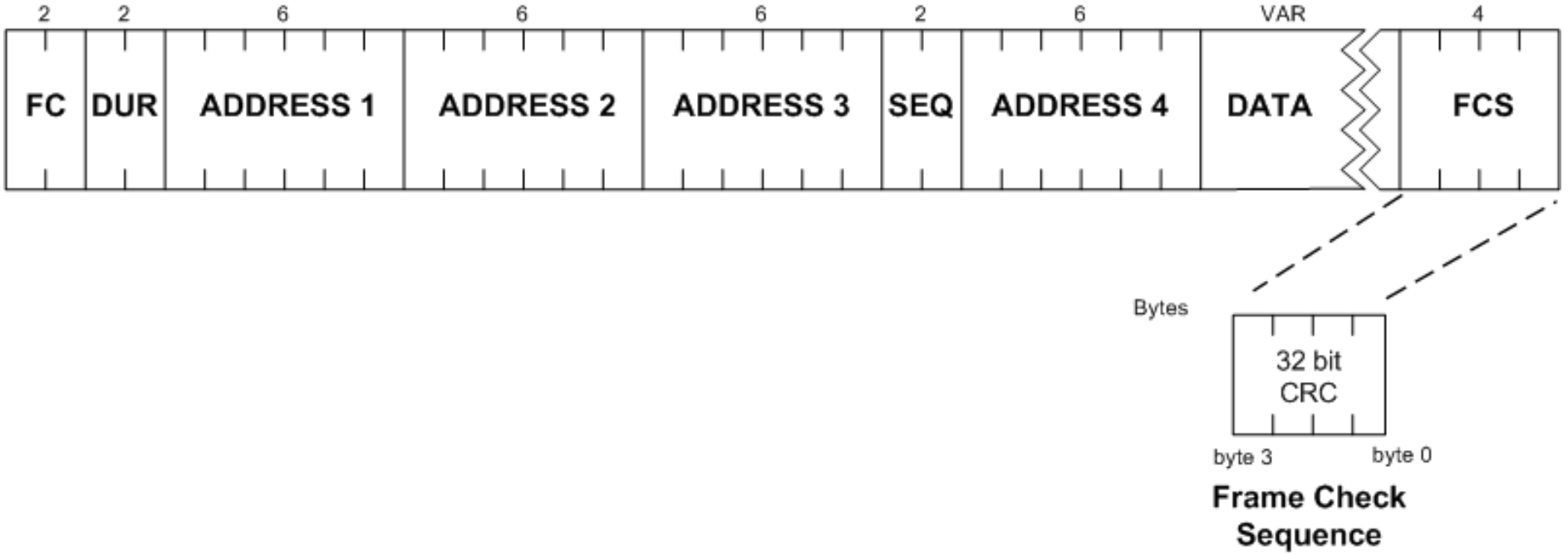


802.11 Sequence Control field



Sequence Control Fields

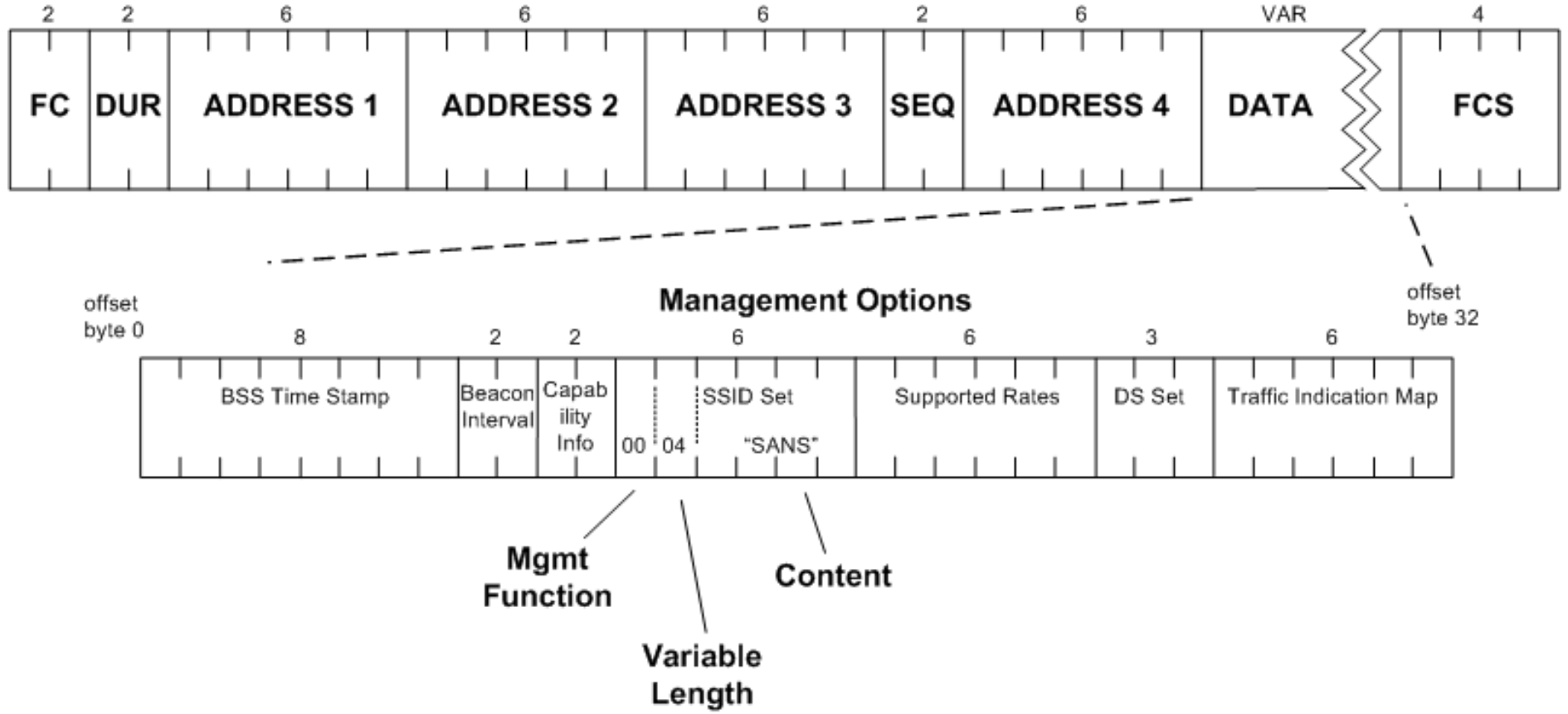
802.11 Frame Check Sequence



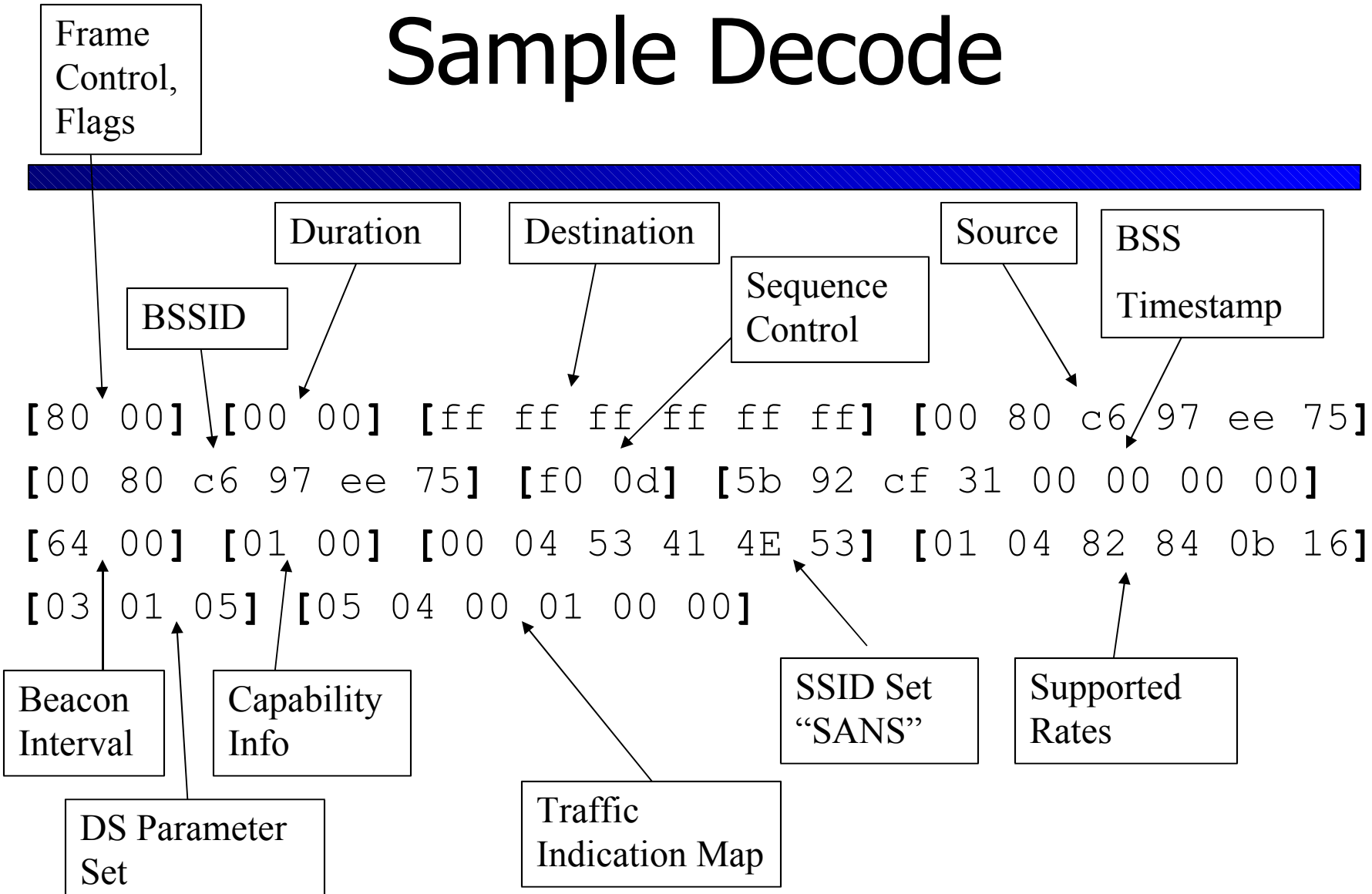
802.11 Management Frames

- Uses Frame Type 00
- Sub-type changes with function of frame
- Management structure assembled in payload
  - Fixed length parameters, strictly ordered
  - Variable length fields, any order

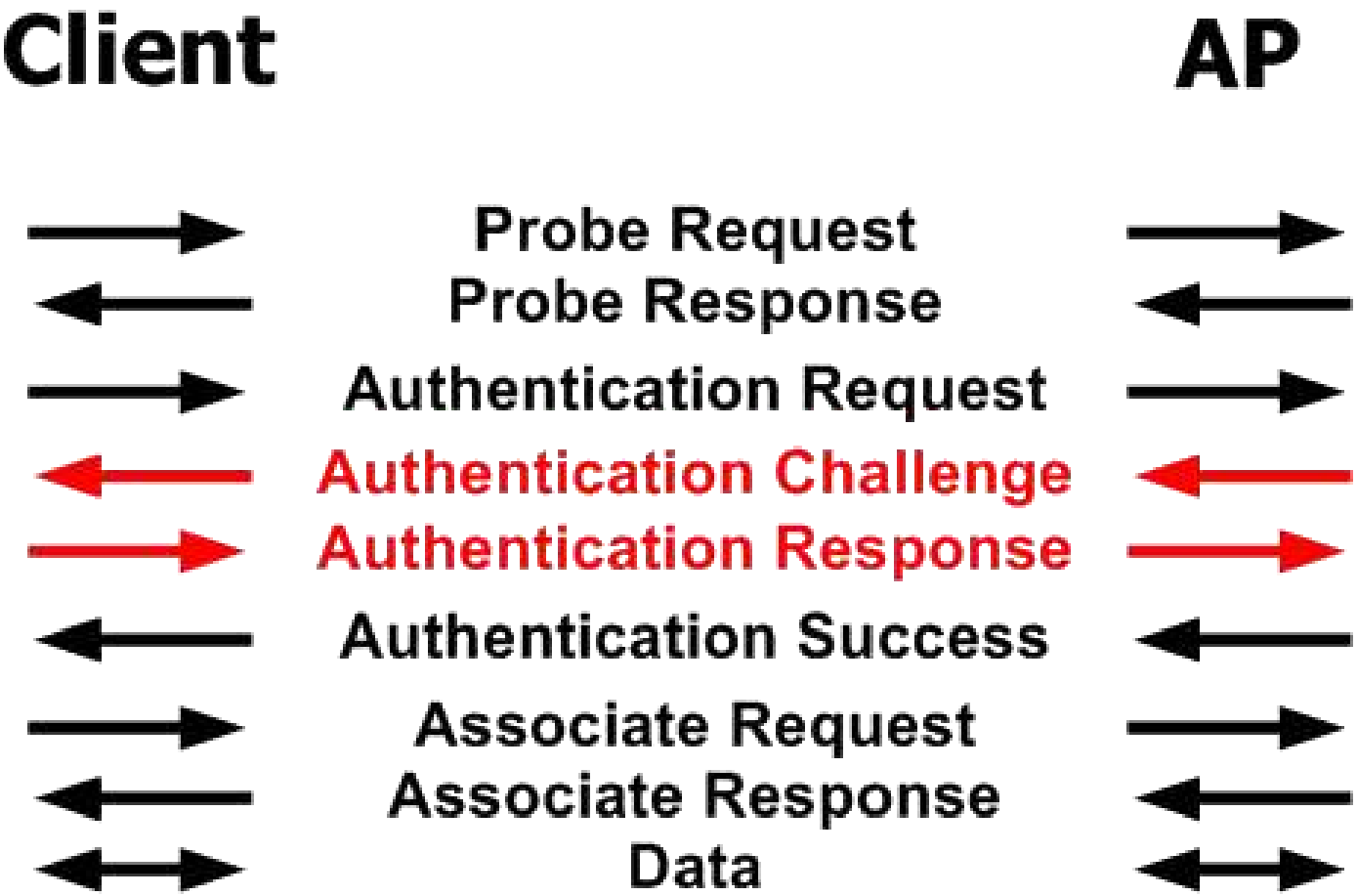
802.11 Management Frames



Sample Decode



Authentication and Association

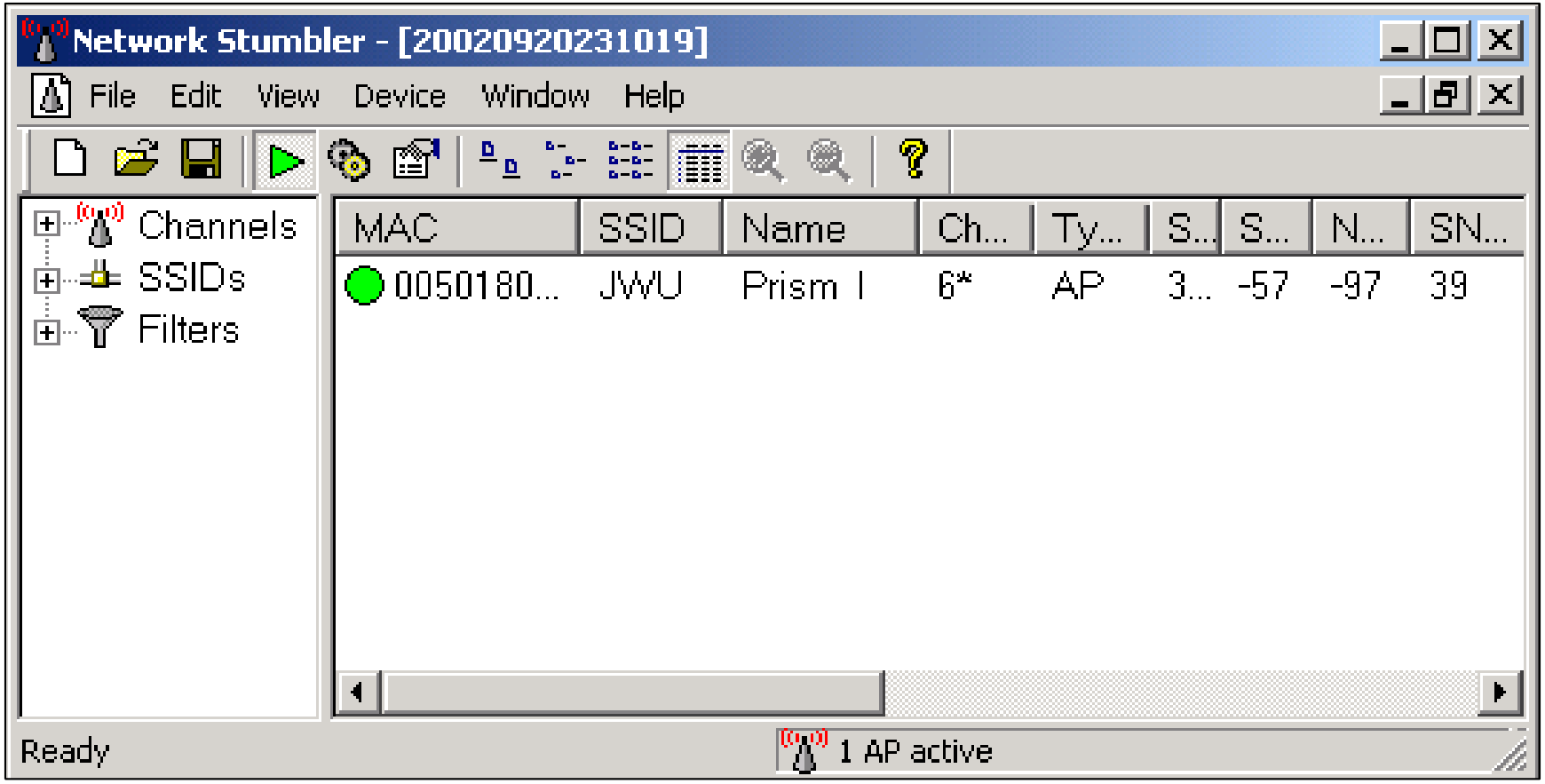


Tools for the Intrusion Analyst

- tcpdump
  - Captures and stores 802.11 frames in libpcap format
- Ethereal
  - Display filters offer flexibility in looking for interesting frames
  - Useful for creating data extracts to narrow analysis
- Kismet
  - Built-in alarms with signature-based analysis, no rules language
- Commercial Tools
  - AirMagnet, AirDefense Rogue Watch suite, WildPackets AiroPeek NX, Network Chemistry Intrusion Prevention
  - "thin" AP vendors including Aruba Networks' Air Monitor
  - Best choice for Windows shops

Attack Tools, Traces and Detection Techniques (the fun stuff)

Identifying NetStumbler



How NetStumbler works

- Uses probe requests
  - Broadcast destination and BSSID
  - Unspecified SSID
  - Channel hopping
- Probe responses returned by AP's
  - Network BSSID
  - May contain WLAN SSID

How NetStumbler works

- After discovering an AP
  - Query sent for "nickname" information
  - AP may respond with hostname information
  - Consistent protocol OUI/PID
  - Payload string identification

NetStumbler Trace

NetStumbler sends a probe request:

00:02:2d:52:cb:27 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Probe Request

NetStumbler receives a probe response:

00:40:96:41:5b:44 -> 00:02:2d:52:cb:27 IEEE 802.11 Probe Response

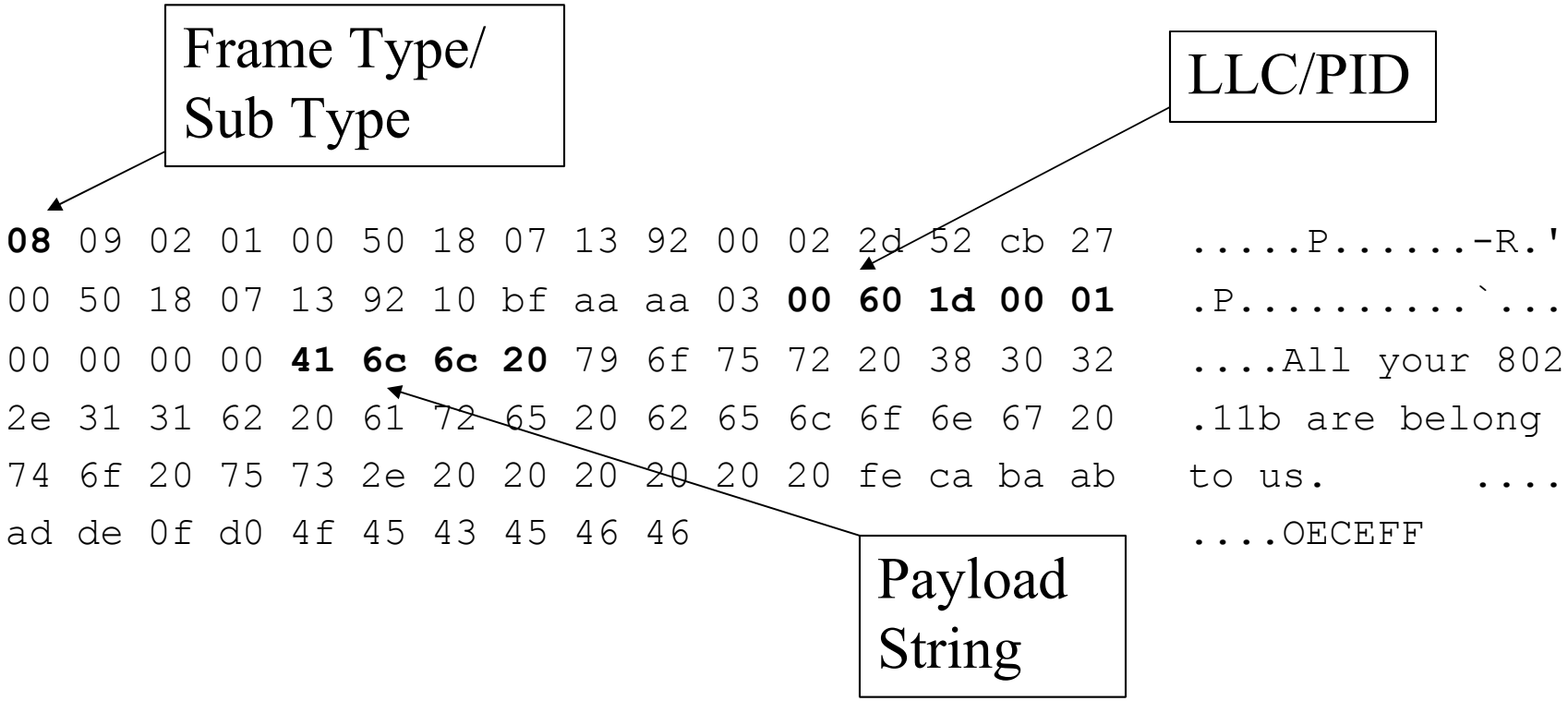
NetStumbler sends a data request for nickname information:

00:02:2d:52:cb:27 -> 00:40:96:41:5b:44 LLC U, func = UI; SNAP, OUI 0x00601D (Unknown), PID 0x0001

NetStumbler Identifier String

- Different versions of NetStumbler use different strings in nickname probes
  - 3.2.0 “Flurble gronk bloopit, bnip Frundletrune”
  - 3.2.1 “All your 802.11b are belong to us”
  - 3.2.3, 3.3.0 “ intentionally blank”

NetStumbler Rule



(wlan.fc.type_subtype eq 32 and llc.oui eq 0x00601d and llc.pid eq 0x0001) and (data [4:4] eq 41:6c:6c:20 or data[4:4] eq 6c:46:72:75 or data[4:4] eq 20:20:20:20)

Identifying FakeAP

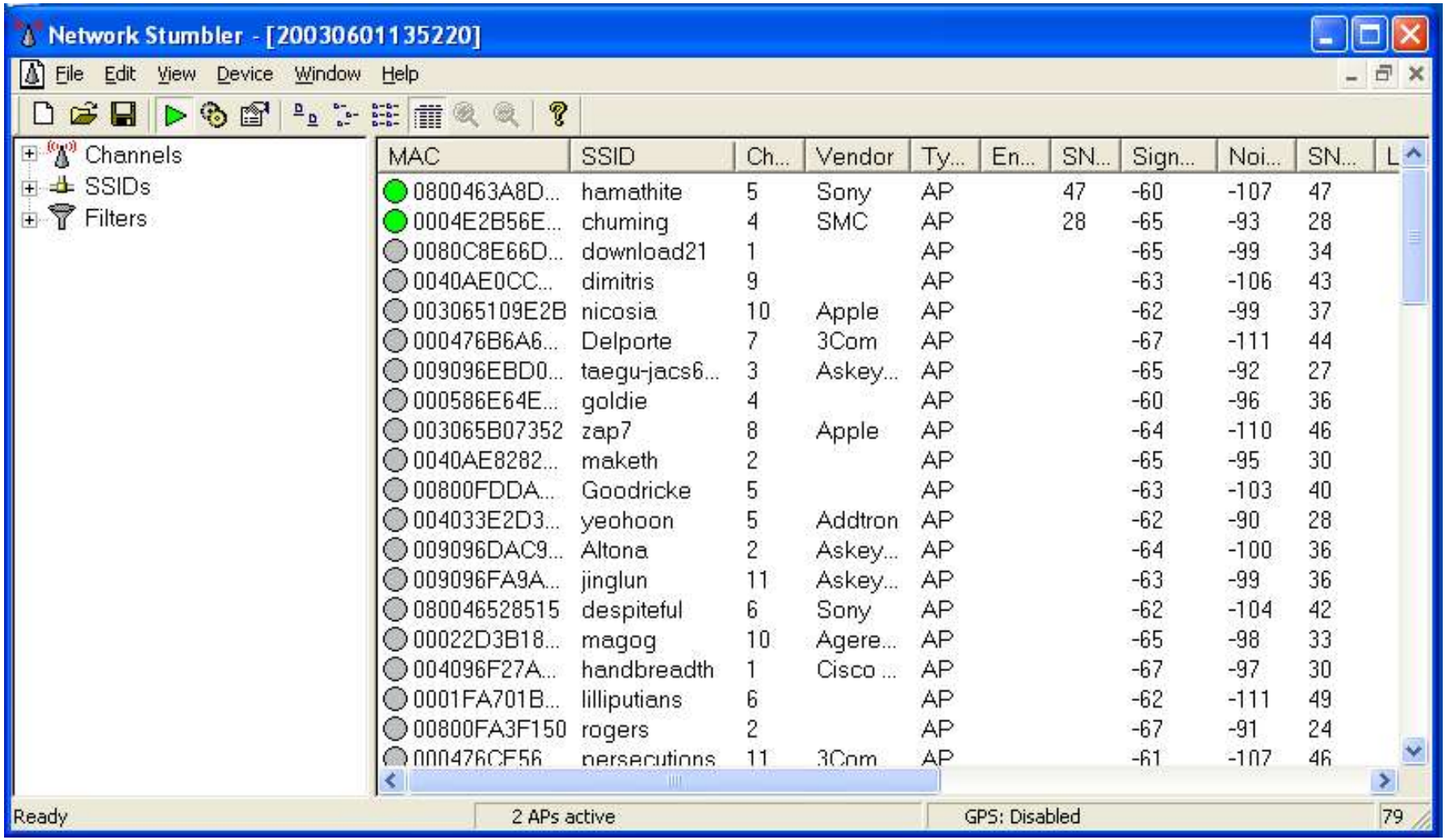
FakeAP Details

- "If one AP is good, 53,000 must be better"
- Uses a dictionary word list to falsely advertise available wireless networks
- Fools discovery tools into reporting multiple network as found
- Not specifically an attack tool
- www.blackalchemy.to

How FakeAP Works

- Uses HostAP Prism2 drivers in "Master" mode
- Cycles through dictionary words, changing the SSID, MAC address, TX power, etc.
- Uses Perl and system() calls to iwconfig
- Can adjust power level, channel and encryption for additional uniqueness qualities
- NetStumbler, Kismet and others report all SSID's as unique wireless networks

What NetStumbler Sees



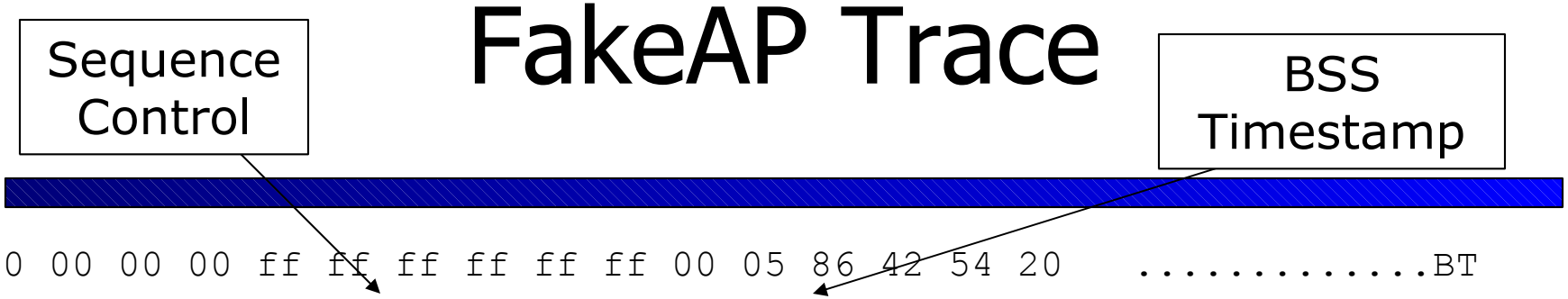
FakeAP Code Extract

```
for ( my $i = 0 ; ; $i++ ) {
 my $essid   = $essid_opt   || gen_essid();
 my $channel = $channel_opt || gen_channel();
 my $mac     = $mac_opt     || gen_mac();

 system($IWCONFIG, $interface_opt, "ESSID", $essid);
 system($IWCONFIG, $interface_opt, "channel", $channel);
 system($IFCONFIG, $interface_opt, "hw", "ether", $mac);
 Time::HiRes::sleep($sleep);
}
```

FakeAP Analysis

- All settings available are set to be unique
  - Tricks wardriver into thinking the AP is real
- Some portions of the 802.11 frame header cannot be set with traditional driver software
  - Real-time functions of 802.11 framing set in card firmware
  - Sequence control field, BSS timestamp
- IDS analyst can use these fields to detect anomalies in traffic

FakeAP Trace




```
80 00 00 00 ff ff ff ff ff ff 00 05 86 42 54 20   ..............BT
00 05 86 42 54 20 50 80 c6 91 01 00 00 00 00 00   ...BT P..a......
64 00 01 00 00 06 63 6f 67 6e 61 63 01 04 82 84   d.....cognac....

80 00 00 00 ff ff ff ff ff ff 00 02 b3 cb 28 64   ..............(d
00 02 b3 cb 28 64 60 80 8b 91 01 00 00 00 00 00   ....(d`.........
64 00 01 00 00 05 6d 61 63 72 6f 01 04 82 84 0b   d.....macro.....

80 00 00 00 ff ff ff ff ff ff 00 02 a5 a5 75 a5   ..............u.
00 02 a5 a5 75 a5 80 80 a2 91 01 00 00 00 00 00   ....u...........
64 00 01 00 00 09 62 72 65 61 74 68 69 6e 67 01   d.....breathing.
```

A distinct pattern! Seq#'s 2053, 2054, 2056
BSS Timestamps always ~ 1/10th of a second

FakeAP Rule

- Watching sequence numbers is memory-intensive
- BSS timestamp < .5 seconds (0x03d070 usec)
  - May false-positive on an AP that has just restarted
  - Can be manually correlated with other traffic analysis
  - At 10 beacons/second, will generate lots of alerts

```
wlan_mgt.fixed.timestamp < "0x000000000003d070"
```

Identifying AirJack

```
/***************************************************************************
**
** Essid Jack: Proof of concept so people will stop calling an ssid a password
**             and its much more eligant than *cough* brute forcing *cough*...
**
**   Author:  Abaddon, abaddon@802.11ninja.net
**
**   Other Development Stuff:  Xx25,  xx25@leper.org
**
**   Copyright (c) 2002 Abaddon, All Rights Reserved (see license info below).
**
***************
**
**   Theory of Operation: (a little long winded, but no-one's making you read)
**
**          We abuse the lack of authentication of 802.11(b) management frames,
**          particularly the de-authentication management frames, to force the
**          extended service set identifier (ESSID) to be transmitted in the
**          clear...works like so...Bob is connected to the access point
**          The people using this access point have been told that the essid
**          is a password or shared secret and so naturaly they have opted to
**          have these essid's not transmitted in the clear (several AP's let
**          let you do this nowa days)...Problem is, when Bob connected to the
**          AP the first time he had to request to be on a network in the same
**          service set...it does this by declaring the service set it wants
**          to be in, in the SSID element field of its probe request frame...
**          so all we have to do is witness him connect and he'll transmit
**          that in the clear...
**
**          now thats all fine and good, you've probably seen tools like
**          Kismet find masked essid's from time to time and this is exactly
**          how they do it...the problem is, well, im a really lazy person and
--More--(12%)
```

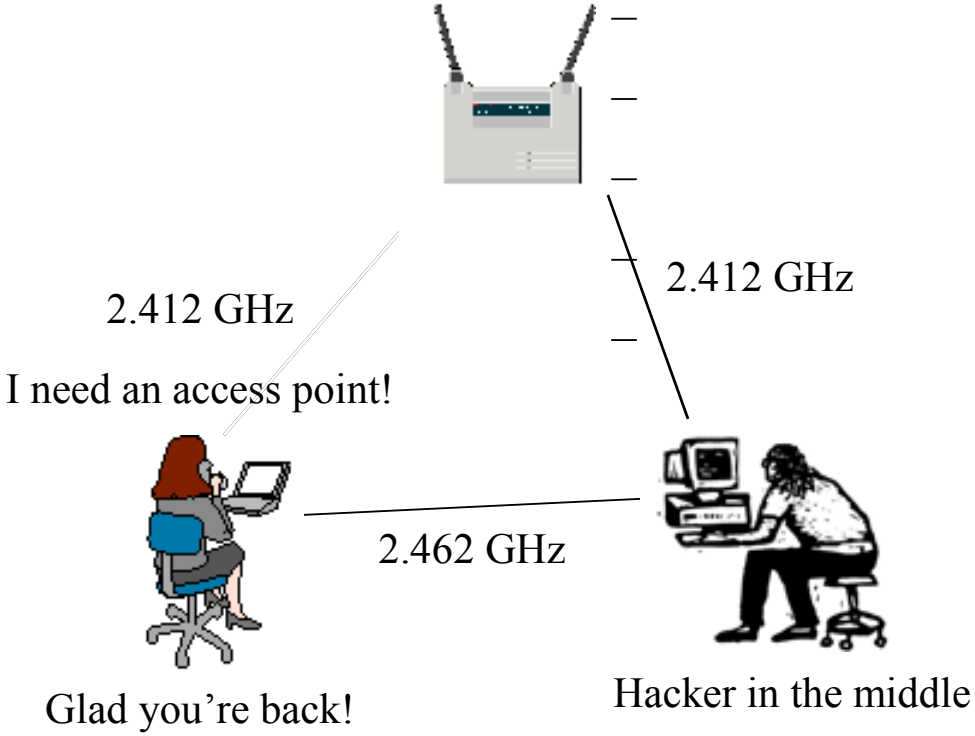
AirJack Details

- Framework for custom 802.11 packet injection
  - Drivers provide functionality for many different WLAN attack tools
  - Linux only
- Written by Abaddon
- Older versions include several attack tools
- http://802.11ninja.net

AirJack Tools

- WLAN-Jack: Sends deauthenticate frames with spoofed source to DoS target network
- ESSID-Jack: Forces a client to reassociate, forcing disclosure of the SSID
- Fata-Jack: More effective version of WLAN-Jack (contributed by loud-fat-bloke)
- Monkey-Jack: Full MITM attack (why we need PEAP/TTLS)
- Kracker-Jack: IPSec MITM attack

How Monkey-Jack Works



Detecting Monkey-Jack

- Attacker uses two cards to connect to legit AP and impersonate legit AP
- Same MAC address on multiple frequencies
  - Shouldn't happen in practice
- Channel-hopping monitors can identify the same BSSID on multiple frequencies
- Stationary monitors will miss the attack

Monkey-Jack Trace

Reported by TCPDump:

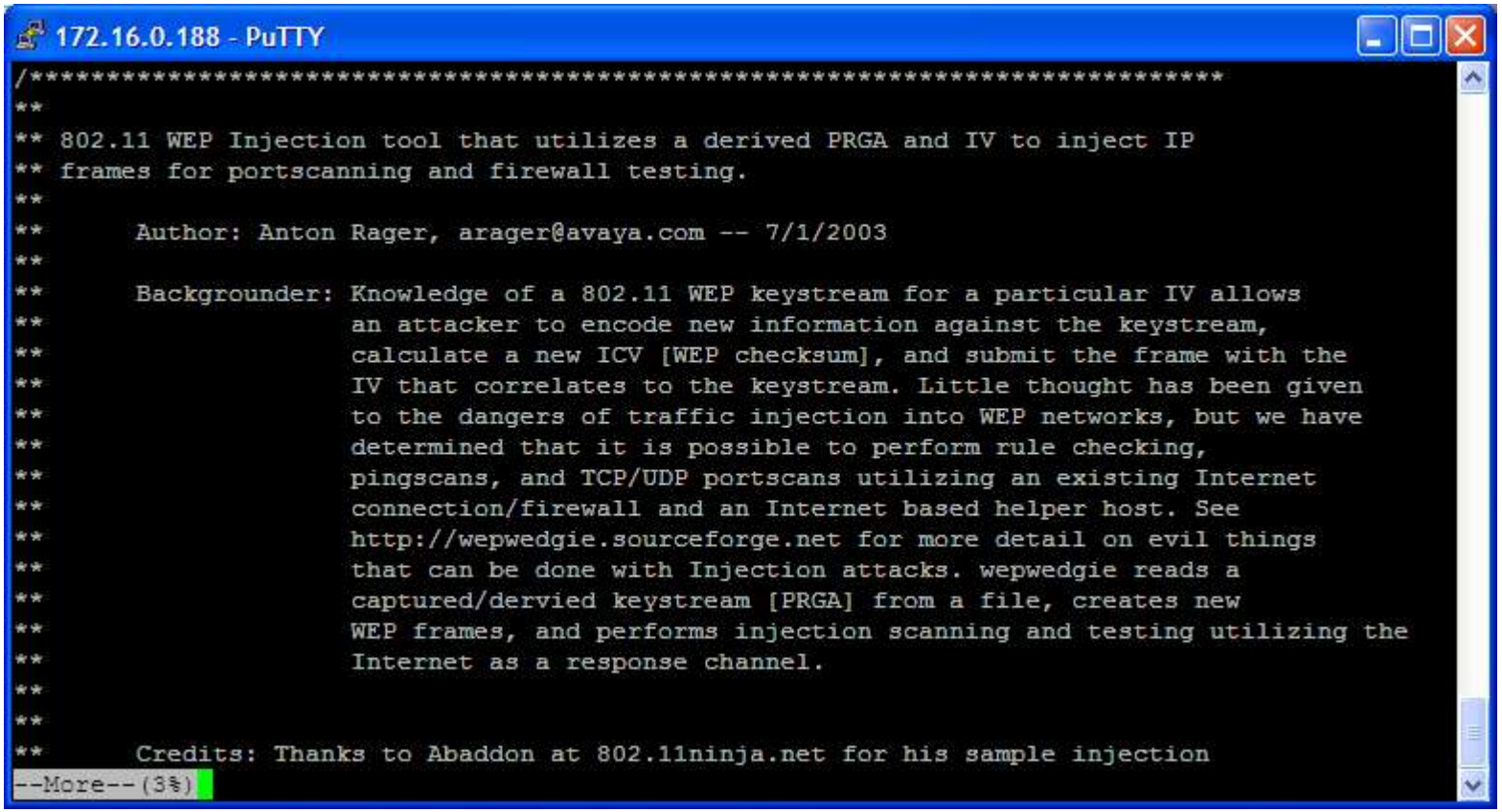
```
BSSID:00:40:96:58:20:58 DA:Broadcast SA:00:40:96:58:20:58
   Beacon (tsunami)  ESS CH: 1, PRIVACY
BSSID:00:40:96:58:20:58 DA:00:07:0e:b8:cf:26
   SA:00:40:96:58:20:58 DeAuthentication: Previous
   authentication no longer valid
BSSID:00:40:96:58:20:58 DA:Broadcast SA:00:40:96:58:20:58
   Beacon (tsunami)  ESS CH: 11, PRIVACY
BSSID:00:40:96:58:20:58 DA:Broadcast SA:00:40:96:58:20:58
   Beacon (tsunami)  ESS CH: 1, PRIVACY
BSSID:00:40:96:58:20:58 DA:Broadcast SA:00:40:96:58:20:58
   Beacon (tsunami)  ESS CH: 1, PRIVACY
BSSID:00:40:96:58:20:58 DA:Broadcast SA:00:40:96:58:20:58
   Beacon (tsunami)  ESS CH: 11, PRIVACY
BSSID:00:40:96:58:20:58 DA:00:40:96:55:75:f5
   SA:00:07:0e:b8:cf:26 Authentication (Reserved)-1: Succesful
```

The authenticate frame transmitted on channel 11 to the Monkey-Jack AP

Monkey-Jack Rule

- No rule functions that span multiple frames in Ethereal
- Multiple deauthenticate frames are a clue
  - Potential deauthenticate flood
- Kismet will report anomalous activity
  - Identifies the same BSSID on multiple channels (advantage of channel-hopping)
  - Not specifically as AirJack
- Stationary channel IDS sensors won't detect this kind of activity

Identifying WEPWedgie



Simplified WEP Process

- RC4 requires two inputs
  - Size of clear-text
  - WEP Key (40/104 bit key + 24 bit IV/nonce)
- Generates PRGA
- WEP XOR's PRGA with clear-text
  - Generates cipher-text
- Receiving station uses cipher-text size, IV and 40/104 bit key to produce same PRGA
  - XOR produces clear-text
- Attacker doesn't have 40/104 bit key, cannot get clear-text

WEP Authentication Process

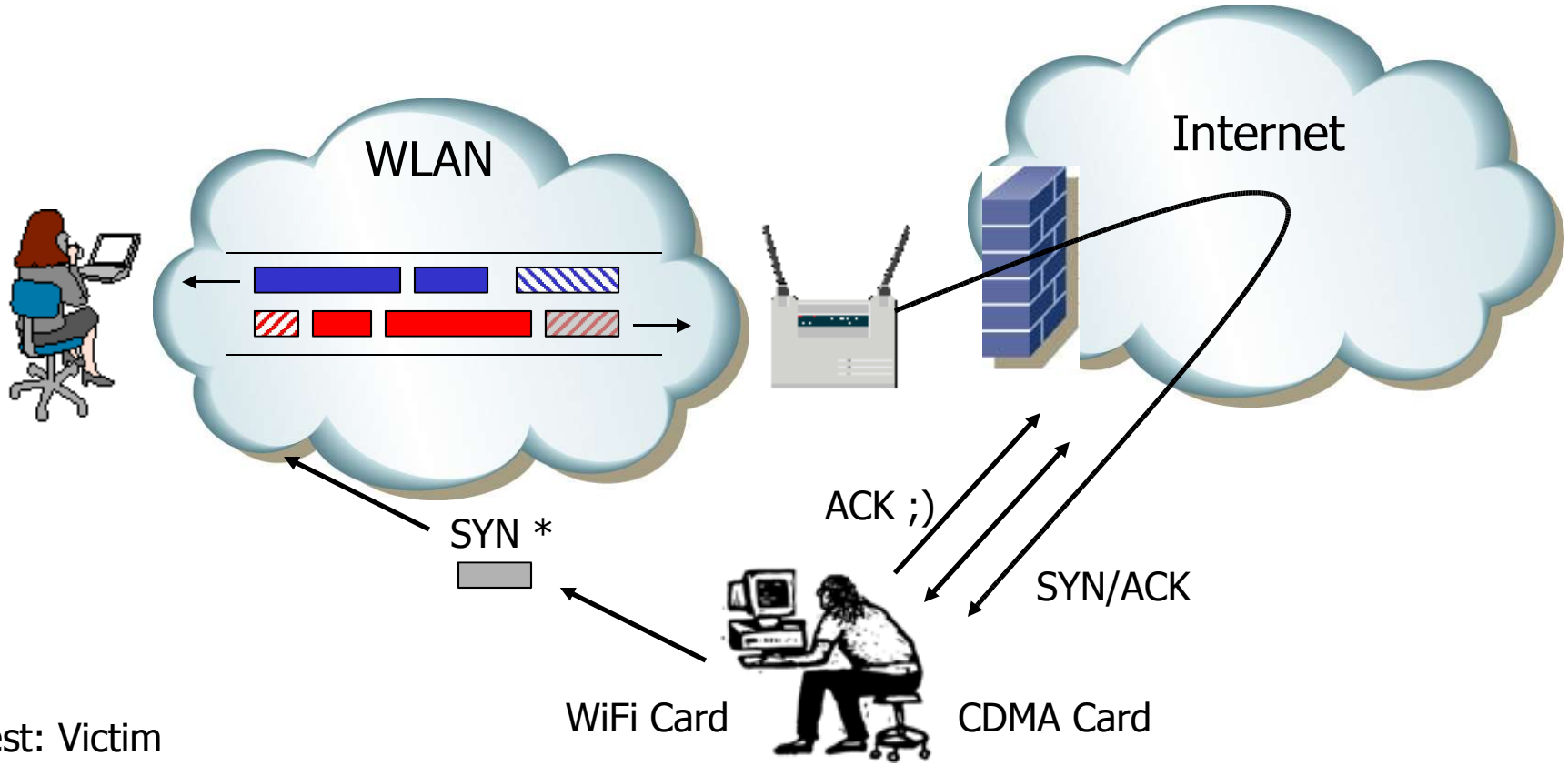
- WEP authentication uses challenge/response text to verify key
  - AP sends challenge text (128 bytes)
  - STA sends response cipher-text
  - AP validates cipher-text, STA is authenticated
- Attacker can monitor this conversation
  - Sees challenge-text
  - Sees cipher-text
  - Sees IV
- Attacker can derive valid PRGA
  - XOR's challenge and cipher text to get PRGA

[Plain XOR PRGA = Cipher] [Cipher XOR PRGA = Plain]

PRGA – So What?

- Attacker can inject arbitrary packets into the network
  - Must be < 128 bytes
- AP verifies cipher-text as valid
  - AP sends packet on its way into the wired network
- Attacker can use multiple mechanisms to determine known-plaintext
  - Not just basic WEP authentication

PRGA Injection Visually



* Dest: Victim
Source: CDMA Card IP

PRGA Injection - Applications

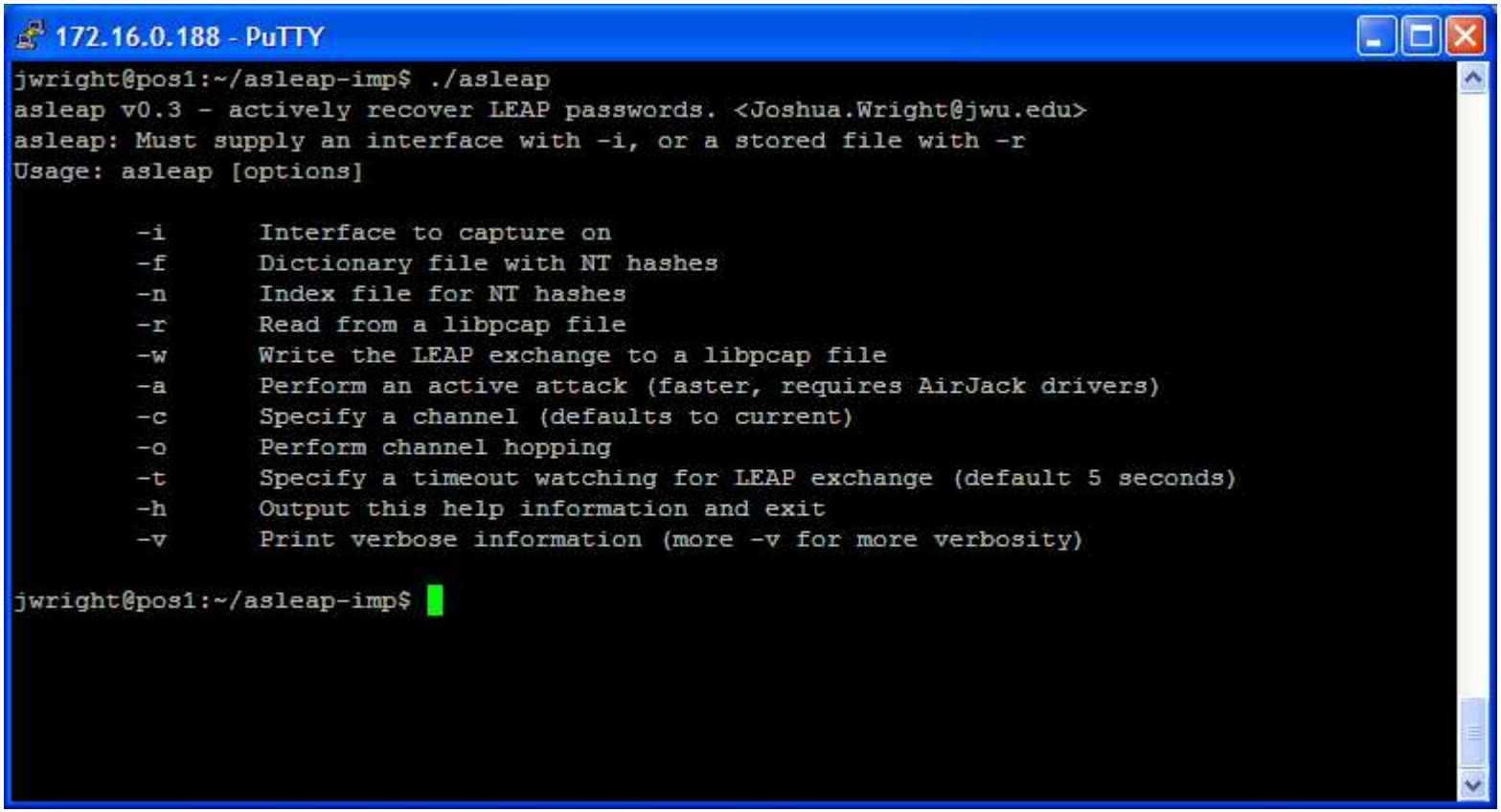
- Accelerate WEP cracking by generating more traffic
- Port-scan hosts behind WEP'd network
- Bypass stateful firewall rules at egress points

All without knowing the WEP key!

Identifying PRGA Injection

- Attacker can inject arbitrary plain-text
  - FCS and packet payload change for each packet
- Sequence numbers are out-of-order
  - Could identify anomalous activity, but doesn't identify the severity or impact of the attack
- Consistent IV from the attacker
  - Attacker must continue to reuse the IV captured during authentication of client
  - Changing the IV would invalidate PRGA
- Kismet to add this detect functionality (in the near future)

Identifying asleap



How asleap works

- Actively disassociates users to force reauthentication
  - Spoofs deauth packet from AP
- Collects MS-CHAPv2 credentials used for LEAP authentication
- Leverages pre-computed dictionary of MD4 hashes to identify the user's password
- Displays username and password

asleap Analysis

- Packets from attacker have forged MAC and BSSID addresses
- Attacker cannot modify parameters handled in firmware of his 802.11 card
  - Sequence control field handled by firmware
  - Cannot be set by AirJack drivers (as it exists today)
- Misaligned sequence number pattern will indicate forged frames

asleap Trace (active mode)

```
00:40:96:55:75:f5 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0000  80 00 00 00 ff ff ff ff ff ff 00 40 96 55 75 f5   ...........@.Uu.
0010  00 40 96 55 75 f5 00 3c a5 d1 e2 ef 4d 00 00 00   .@.Uu..<....M...

00:07:0e:b8:cf:26 -> 00:00:0c:07:ac:01 IEEE 802.11 Data

00:40:96:55:75:f5 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0000  80 00 00 00 ff ff ff ff ff ff 00 40 96 55 75 f5   ...........@.Uu.
0010  00 40 96 55 75 f5 00 4c ac e1 f0 ef 4d 00 00 00   .@.Uu..<....M...

00:40:96:55:75:f5 -> 00:07:0e:b8:cf:26 IEEE 802.11 Deauthentication
0000  c0 00 00 00 00 07 0e b8 cf 26 00 40 96 55 75 f5   .........&.@.Uu.
0010  00 40 96 55 75 f5 0a 10 02 00 00 00               .@.Uu.......
                          <frames deleted for brevity>
00:07:0e:b8:cf:26 -> 00:40:96:55:75:f5 EAP Request, EAP-Cisco Wireless (LEAP)
```

Sequence numbers are out of order with consistent source MAC's. Deauthenticate frame is spoofed. (195, 196, 416)

Using RSSI Anomaly Analysis

- Some wireless cards can report additional frame information
  - Signal level, noise level for each packet
- IDS can monitor signal strength for each unique source
- Sudden changes in signal strength (reduction or increase) may indicate a spoofed source
  - Attacker has different physical proximity to the AP than the legitimate user
  - Can be used to generalize location of attacker

Extended Capture Information

- Collect signal, noise data keyed on STA source addresses
- Use trend analysis to establish a "normalized" operating environment

```
typedef struct p80211_avshdr {
    <snip>
    UINT32    channel;
    UINT32    datarate;
    UINT32    rssi_type;
    INT32     rssi_signal;
    INT32     rssi_noise;
    <snip>
} p80211_avshdr_t;

typedef struct p80211_hdr {
    UINT32    frame_control;
    UINT32    duration;
    UCHAR     dstaddr[6];
    UCHAR     srcaddr[6];
    UCHAR     bssid[6];
    <snip>
} p80211_hdr_t;
```

RSSI Data Trace

Signal Strength

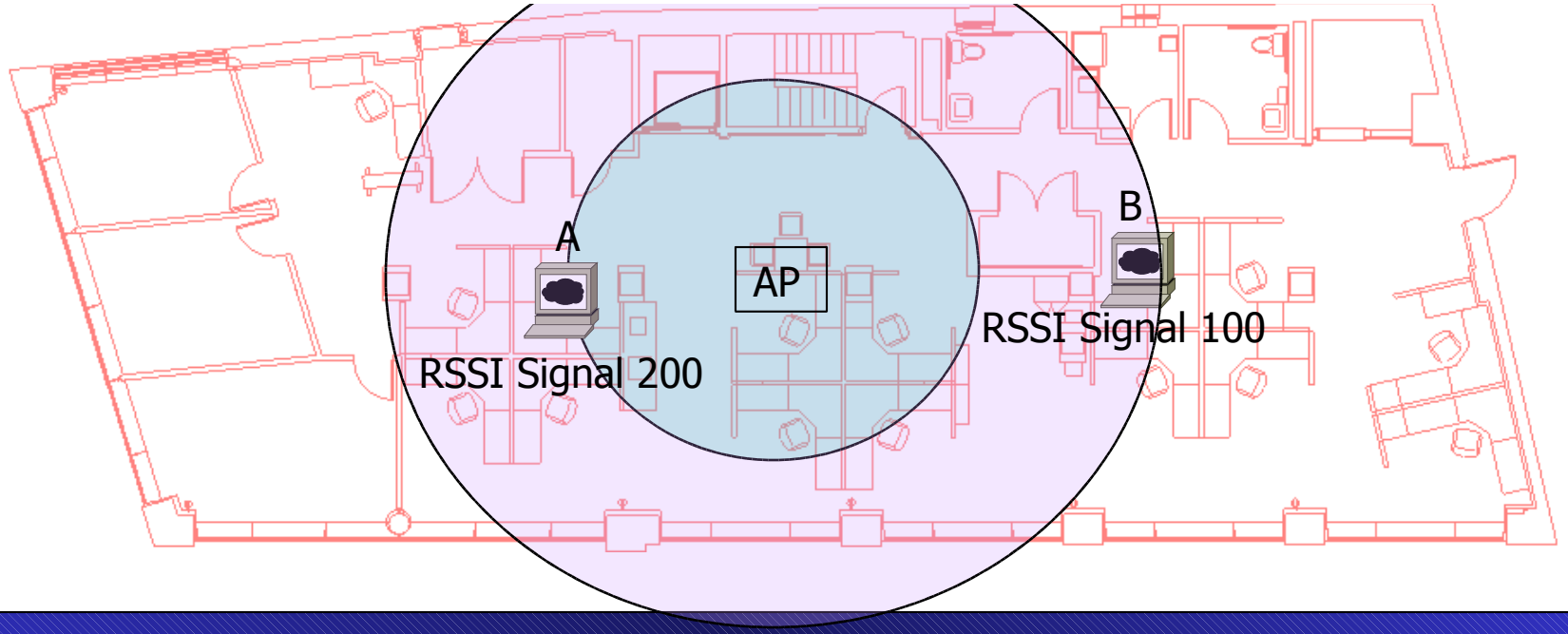
Noise Level

```
00:40:96:55:75:f5 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0000  80 21 10 01 00 00 00 40 00 00 00 00 4c b3 f6 8c   .!.....@....L...
0010  00 00 00 00 00 0c a9 42 00 00 00 04 00 00 00 01   .......B........
0020  00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 03   ................
0030  00 00 00 cf 00 00 00 9f 00 00 00 00 00 00 00 01   ................

00:40:96:55:75:f5 -> 00:07:0e:b8:cf:26 IEEE 802.11 Deauthentication
0000  80 21 10 01 00 00 00 40 00 00 00 00 4c b4 b0 2d   .!.....@....L..-
0010  00 00 00 00 00 0c a9 45 00 00 00 04 00 00 00 01   .......E........
0020  00 00 00 6e 00 00 00 00 00 00 00 00 00 00 00 03   ...n............
0030  00 00 00 65 00 00 00 9f 00 00 00 00 00 00 00 01   ...e............

00:40:96:55:75:f5 -> ff:ff:ff:ff:ff:ff IEEE 802.11 Beacon frame
0000  80 21 10 01 00 00 00 40 00 00 00 00 4c b5 5f 17   .!.....@....L._.
0010  00 00 00 00 00 0c a9 4a 00 00 00 04 00 00 00 01   .......J........
0020  00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 03   ................
0030  00 00 00 d2 00 00 00 9e 00 00 00 00 00 00 00 01   ................
```

Source MAC and noise level are consistent, signal strength changes from 207 to 101 to 210 (0xcf, 0x65, 0xd2) – frames trimmed for brevity

Identifying Location

- WIDS monitor can identify location of attacker
- Uses relative knowledge of operating environment (other stations)



Summary

- The 802.11 specification is complex, opening up potential misinterpretations that can be abused
- IDS analysts can use multiple methods to detect attacks against 802.11 networks
  - Signature analysis
  - Anomalous behavior patterns
  - Sequence number analysis
  - RSSI data correlation
- Ethereal, tcpdump and Kismet are your friends

Links

- Kismet – WLAN Discovery, Intrusion Alerting, Rogue AP detection
  - http://www.kismetwireless.net/
- FakeAP – Tricking NetStumbler
  - http://www.blackalchemy.to/project/fakeap/
- NetStumbler – Windows WLAN Discovery Application
  - http://www.netstumbler.com/
- AirJack – Advanced 802.11 injection framework, tools
  - http://802.11ninja.net/
- WEPWedgie
  - http://www.sourceforge.net/projects/wepwedgie/

Questions?

"tcp[13] & 0x01 != 0"

Please don't forget to return your evaluation forms! Thank you!

Joshua Wright / jwright@sans.org
Home Page: http://home.jwu.edu/jwright/
PGP Key: http://home.jwu.edu/jwright/pgp.htm

Other Related Papers by the author:

"Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection"
http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf

"Detecting Wireless LAN MAC Address Spoofing"
http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf