



Interested in learning
more about security?

SANS Institute InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Understanding Wireless Attacks and Detection

This paper introduces wireless attacks from a OSI layer 2 perspective and attempts to understand how wireless attacks can be detected by looking at wireless frames at these layers.

Copyright SANS Institute
Author Retains Full Rights

AD

Veriato

Unmatched visibility into the computer
activity of employees and contractors



Try Now

Understanding Wireless attacks & detection

GIAC Security Essentials
Certification (GSEC)
Practical Assignment
Version 1.4c

Option 1 - Research on Topics
in Information Security

Submitted by: Christopher Low 13 April 2005

Paper Abstract: This paper introduces wireless attacks from a OSI layer 2 perspective and attempts to understand how wireless attacks can be detected by looking at wireless frames at these layers.

Current wireless IDS detection techniques and technologies are then introduced and evaluated for its effectiveness in light of these attacks.

© SANS Institute 2005, Author retains full rights.

Table of Contents

Executive Summary	1
Introduction to 802.11 wireless frames	2
Capturing wireless packets	2
Dissecting wireless packets	4
Wireless Attacks	8
Probing & Network Discovery	8
Surveillance	9
DOS attacks	10
Impersonation	11
Man in the middle and Rouge AP	12
Wireless Attacks Detection Techniques	13
Access Point Monitoring	13
Wireless Client Monitoring	13
General Wireless Traffic Monitoring	14
Wireless IDS	15
snort-wireless	15
WIDZ	15
AirIDS	15
Conclusion	16
References	17

List of Figures

Figure 1 - Wireless packets sniffing	2
Figure 2 - Opening Kismet capture using ethereal	3
Figure 3 - Beacon frame	4
Figure 4 - Select all packets from a particular BSSID	4
Figure 5 - Frame sequence number	5
Figure 6 - Capturing the SSID value	5
Figure 7 - SSID is cloaked	6
Figure 8 - Normal wireless client association traffic	6
Figure 9 - A wireless data packet	7
Figure 10 - A WEP encrypted packet	7
Figure 11 - Kismet	9
Figure 12 - Aircrack used to crack WEP key	10
Figure 13 - MAC spoofing in linux	11
Figure 14 - MAC spoofing in Windows	12
Figure 15 - Man in the middle	13

Christopher Low	Understanding wireless attacks and detection
-----------------	--

Executive Summary

Productivity gain, ease of use and deployment, work mobility are all reasons why wireless technologies have not only infiltrated the homes of many but also enterprises. Enterprise wireless deployments have become very prevalent.

The functionalities and convenience brought about by wireless technologies have also introduced risks into these traditionally “closed” networks to allow for access from anywhere where the RF signals could reach.

This paper will first start by introducing the topic of wireless technology (specifically 802.11 technology) from the wireless frame layer (OSI layer 2) perspective. We’ll be looking at wireless frames and understanding how they can be captured, dissected and analyzed.

We’ll then proceed to look at different types of wireless attacks and how those attacks can be recognized at the wireless frame layer.

Wireless IDS technologies are then introduced and we’ll be looking briefly at how effective and comprehensive they are when it comes to detecting for such forms of wireless attack.

Introduction to 802.11 wireless frames

The best way to understand how any type of network protocol work is to look at it from the packet level. I'll first be describing how we can capture wireless frames that are flying through the air and then talk about dissecting and analyzing these frames for purposes of understanding how these packets look like under normal traffic conditions.

Capturing wireless packets

Unlike traditional wired networks where network packets are transmitted along physical wires, wireless technologies use the air as the physical media when sending and receiving data packets.

Air being the physical media where wireless packets traverse, opens up whole new opportunities for anyone within the vicinity (either of the wireless client or the wireless access point) with the right devices and software to be able to capture those packets.

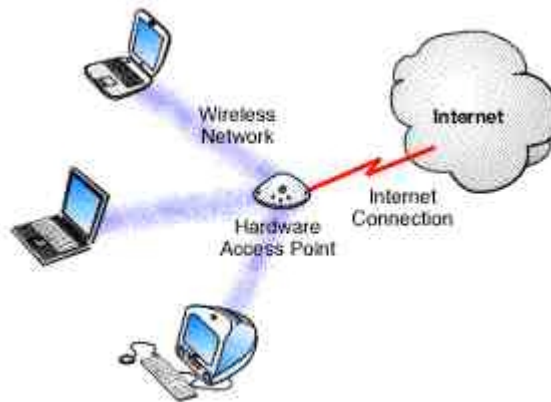


Figure 1 - Wireless packets sniffing

In order to capture wireless packets, the sniffing station will have to be equipped with the appropriate hardware and software.

The hardware required will depend on the wireless network being targeted and they come in the form of a wireless network interface card (CF, PCI, USB, PCMCIA based & on-board wireless NICs/chipsets are available). Refer to http://www.linux-wlan.org/docs/wlan_adapters.html.gz for a list of wireless NICs as well as their corresponding chipsets.

Depending on the device drivers as well as the capability of each chipset, they might differ in support when it comes to the software that we're about to introduce for purposes of sniffing these traffic.

The software, which we'll be using for purposes of wireless traffic sniffing in this research, is Kismet (<http://www.kismetwireless.net>). And to assist in our analysis of the sniffed packets, we'll be using ethereal (<http://www.ethereal.com>).

Kismet is an 802.11 layer 2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card that supports raw monitoring (rfmon) mode. It can sniff 802.11b, 802.11a, and 802.11g traffic.

Kismet can be installed very easily on any unix based operating system and can also be executed in Windows running CYGWIN (<http://www.cygwin.com>).

Under normal circumstances when your wireless NIC are sending and receiving wirelessly, they're placed into a mode called managed mode, and in this mode, the wireless NIC will not pick up any wireless packets which are not destined for it, thus defeating the purposes of being a sniffing station. In order for the wireless NIC to pick up all packets regardless of who the packet is for, the NIC will have to be placed in rfmon mode. The problem with running Kismet in Windows boils down to the difficulty in getting the wireless NIC into rfmon mode as most publicly available Win32 drivers don't support the rfmon mode by default.

Packets captured by Kismet can be saved into pcap files, which are then analyzed by ethereal by opening those files in an offline mode.

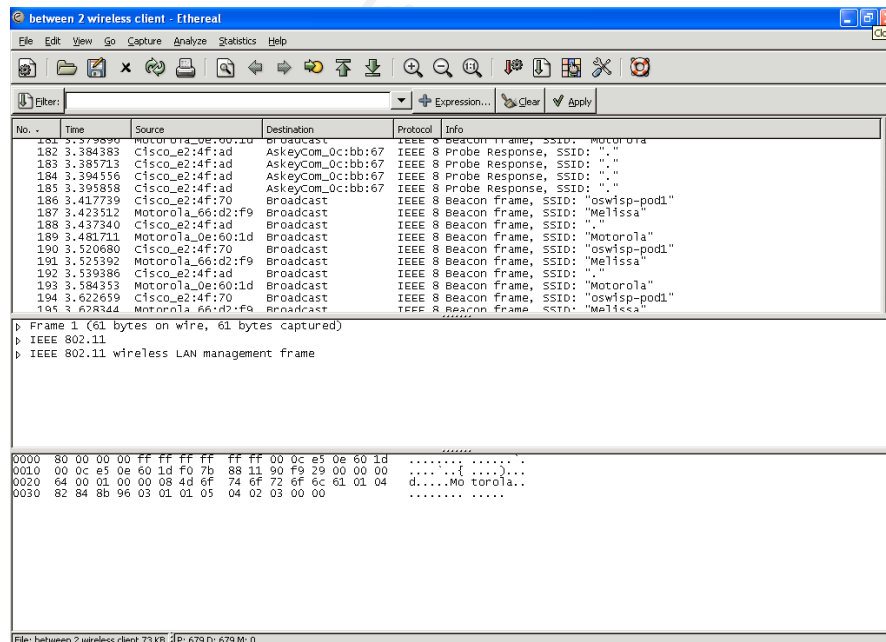


Figure 2 - Opening Kismet capture using ethereal

Dissecting wireless packets

Let's now extract some common wireless packet types and observe how wireless communication is made possible.

No. -	Time	Source	Destination	Protocol	Info
187	3.423512	Motorola_66:d2:f9	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "Melissa"
188	3.437340	Cisco_e2:4f:ad	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: ""
189	3.481711	Motorola_0e:60:1d	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "Motorola"


```

Frame 187 (60 bytes on wire, 60 bytes captured)
  Arrival Time: Apr  8, 2005 18:18:22.785079000
  Time delta from previous packet: 0.005773000 seconds
  Time since reference or first frame: 3.423512000 seconds
  Frame Number: 187
  Packet Length: 60 bytes
  Capture Length: 60 bytes
  IEEE 802.11
    Type/Subtype: Beacon frame (8)
    Frame Control: 0x0080 (Normal)
    Duration: 0
    Destination address: ff:ff:ff:ff:ff:ff (Broadcast)
    Source address: 00:0b:06:66:d2:f9 (Motorola_66:d2:f9)
    BSS Id: 00:0b:06:66:d2:f9 (Motorola_66:d2:f9)
    Fragment number: 0
    Sequence number: 982
  
```

Figure 3 - Beacon frame

One of the most common wireless frames you'll see whenever you do any kind of wireless sniffing would be beacon frames. Figure 3 shows an example of a captured beacon frame.

A beacon frame is a packet send by a wireless access point (on a regular basis) in a wireless infrastructure mode operation to allow wireless clients within the vicinity to detect the Station Set Identifier (SSID) of the wireless networks. SSID defines the name of the wireless network that all the wireless clients associate with.

The first address field in the beacon frame is the **destination address**. This field has a value of "ff:ff:ff:ff:ff:ff", this value indicates that the packet is to be send to all stations. The third address field is the **BSS ID** (Basic Station System ID) field which contains the MAC address for the wireless side of the access point.

Using the filter mechanism in ethereal, we can effectively select packets belonging to this wireless network by simply keying in the following filter string "wlan.bssid==**bssid value**" in the filter field in ethereal (see figure 4 below).

The screenshot shows the Wireshark interface with the filter field set to "wlan.bssid == 00:0b:06:66:d2:f9". The packet list pane shows three captured packets, all of which are IEEE 802.11 Beacon frames from the Motorola_66:d2:f9 source.

No. -	Time	Source	Destination	Protocol	Info
187	3.423512	Motorola_66:d2:f9	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "Melissa"
191	3.525392	Motorola_66:d2:f9	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "Melissa"
195	3.628344	Motorola_66:d2:f9	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "Melissa"

Figure 4 - Select all packets from a particular BSSID

Another field to note in this packet is the **sequence number** field; this field is incremented by one every time the wireless station emits a packet. In figure 5

below, we see that the sequence number for this beacon frame packet is 982. Therefore the next packet that will be emitted by this wireless station would then be 983.

```

Frame Number: 187
Packet Length: 60 bytes
Capture Length: 60 bytes
IEEE 802.11
  Type/Subtype: Beacon frame (8)
  Frame Control: 0x0080 (Normal)
  Duration: 0
  Destination address: ff:ff:ff:ff:ff:ff (Broadcast)
  Source address: 00:0b:06:66:d2:f9 (Motorola_66:d2:f9)
  BSS Id: 00:0b:06:66:d2:f9 (Motorola_66:d2:f9)
  Fragment number: 0
  Sequence number: 982
IEEE 802.11 wireless LAN management frame
  Fixed parameters (12 bytes)
  Tagged parameters (24 bytes)

```

Figure 5 - Frame sequence number

Under the tagged parameters, the SSID of this particular wireless LAN can then be found, "Melissa" in this example. See figure 6 below.

No. -	Time	Source	Destination	Protocol	Info
3	0.044379	Motorola_66:d2:f9	Broadcast	IEEE 802.11	Beacon frame, S

```

Frame 3 (60 bytes on wire, 60 bytes captured)
IEEE 802.11
  IEEE 802.11 wireless LAN management frame
  Fixed parameters (12 bytes)
  Tagged parameters (24 bytes)
    Tag Number: 0 (SSID parameter set)
    Tag length: 7
    Tag interpretation: Melissa
    Tag Number: 1 (Supported Rates)
    Tag length: 4
    Tag interpretation: Supported rates: 1.0(B) 2.0(B) 5.5 11.0 [Mbit/sec]
    Tag Number: 3 (DS Parameter set)

```

Figure 6 - Capturing the SSID value

In figure 7 below, we observe that the SSID value of this particular wireless LAN is cloaked. This can be setup in various Access points to prevent the SSID from being broadcast. This is usually recommended as a best practice so as to prevent casual wardrivers from picking up this network using war driving software.

The image shows a Wireshark packet capture window with a filter set to 'wlan.fc.type==0 and wlan.fc.type_subtype==0x08'. The packet list shows three IEEE 802.11 Beacon frames. The selected packet (No. 931) is expanded to show details: Type/Subtype: Beacon frame (8), Frame Control: 0x0080 (Normal), Duration: 0, Destination address: ff:ff:ff:ff:ff:ff (Broadcast), Source address: 00:10:c6:2b:42:4a (Usi_2b:42:4a), BSS Id: 00:10:c6:2b:42:4a (Usi_2b:42:4a), Fragment number: 0, Sequence number: 7. The IEEE 802.11 wireless LAN management frame section is expanded to show Fixed parameters (12 bytes) and Tagged parameters (26 bytes). The Tagged parameters section shows Tag Number: 0 (SSID parameter set), Tag length: 1, and Tag interpretation: Tag Number: 1 (Supported Rates).

No. -	Time	Source	Destination	Protocol	Info
930	265.727749	Usi_2b:42:4a	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "
931	265.830161	Usi_2b:42:4a	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "
932	265.932573	Usi_2b:42:4a	Broadcast	IEEE 802.11	IEEE 802.11 Beacon frame, SSID: "

Figure 7 - SSID is cloaked

Let's now look at how a typical wireless client would connect to the wireless AP. In this example below, a wireless client is trying to connect to a predefined SSID configured and 6 packets that are shown below includes a probe request by the wireless client followed by a probe response, 2 authentication packets (Open Authentication is engaged) as well as 2 association packets. After these 6 packets are exchanged, the wireless client can start sending and receiving packets over this wireless network.

The image shows a Wireshark packet capture window with a filter set to 'wlan.fc.type == 0 and wlan.fc.type_subtype!=0x08'. The packet list shows six IEEE 802.11 packets related to association. The selected packet (No. 1898) is expanded to show details: Type/Subtype: Probe Request, Frame Control: 0x0000 (To DS: 0, From DS: 0, To DS: 0, From DS: 0, Protected: 0, More Fragments: 0), Duration: 0, Destination address: ff:ff:ff:ff:ff:ff (Broadcast), Source address: 192.168.2.52, BSS Id: 192.168.2.52, Fragment number: 0, Sequence number: 1. The IEEE 802.11 wireless LAN management frame section is expanded to show Fixed parameters (12 bytes) and Tagged parameters (26 bytes). The Tagged parameters section shows Tag Number: 0 (SSID parameter set), Tag length: 1, and Tag interpretation: Tag Number: 1 (Supported Rates).

No. -	Time	Source	Destination	Protocol	Info
1898	53.897167	192.168.2.52	Broadcast	IEEE 802.11	IEEE 802.11 Probe Request, SSID: "oswisp-pod2"
1899	53.898148	Cisco_e2:4f:ad	192.168.2.52	IEEE 802.11	IEEE 802.11 Probe Response, SSID: "oswisp-pod2"
1902	53.922135	192.168.2.52	Cisco_e2:4f:ad	IEEE 802.11	IEEE 802.11 Authentication
1904	53.923347	Cisco_e2:4f:ad	192.168.2.52	IEEE 802.11	IEEE 802.11 Authentication
1906	53.924600	192.168.2.52	Cisco_e2:4f:ad	IEEE 802.11	IEEE 802.11 Association Request, SSID: "oswisp-pod2"
1908	53.925750	Cisco_e2:4f:ad	192.168.2.52	IEEE 802.11	IEEE 802.11 Association Response, SSID: Broadcast

Figure 8 - Normal wireless client association traffic

The wireless client can subsequently send and receive packets over the wireless network as depicted in the following diagram.

© SANS Institute

No. -	Time	Source	Destination	Protocol	Info
11021	297.18216	192.168.2.51	192.168.2.255	NBNS	Registration NB STUdENT1<00>
11026	297.35113	192.168.2.51	224.0.0.22	IGMP	v3 Membership Report

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

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

Figure 9 - A wireless data packet

We can see from the above diagram that the wireless client is sending a netbios name service broadcast message over UDP/IP. The Logical Link Control section of the frame contains information about the encapsulated higher protocol (in this example IP protocol).

After looking at a normal packet without any form of encryption, let's look at how a WEP encrypted packet would look like.

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

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

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

Figure 10 - A WEP encrypted packet

You'll notice that the information beyond the IEEE 802.11 header is decoded by ethereal as Data. If you open up the IEEE 802.11 header, you'll notice that there is a WEP parameter section where we can find the 24 bit Initialization vector (IV) used for this particular packet encryption as well as the ICV (used for ensuring data integrity). One other piece of information that is not encrypted and can be inspected by all is the MAC addresses of both communicating parties.

Wireless Attacks

Probing & Network Discovery

Before an attacker is able to attempt any kind of wireless mischief, one of the first activities would be for him to identify the various wireless targets in range. Probing and network discovery type attacks described in this section are amongst the first activities engaged by any attacker

There are primarily 2 main types of probing, active and passive probing. Active probing involves the attacker actively sending probe requests with no SSID configured (very much like a normal wireless client would do) in order to solicit a probe response with SSID information and other information from any access points in range. Active probing cannot detect for access points that are cloaked (configured not to respond to probe requests with no SSID set) or out of range of the attacker's wireless transmission range.

When an attacker engages in passive probing, he is listening on all channels for all wireless packets send and receive without sending even a single packet, thus the detection capability is not limited by his transmission power. But, cloaked APs with no wireless activities during the period of the probe would not be detected.

A good example of a tool that uses active probing is NetStumbler. Kismet (shown in Figure 11) on the other hand is an example of a passive probing tool.

© SANS Institute 2005. All rights reserved.

```

root@fc2:/home/chris
File Edit View Terminal Tabs Help
Network List (SSID)
Name      T W Ch Packts Flags IP Range  Size
<no ssid> A Y 006 85    0.0.0.0  0B
<no ssid> P N --- 2     0.0.0.0  0B
Steven Koh A N 006 323   0.0.0.0  0B
sandyteo  A Y 006 21    0.0.0.0  0B

Info
Ntwrks 5
Pckets 1894
Cryptd 193
Weak 0
Noise 0
Discrd 0
Pkts/s 0

orinoc
Ch: 3

Elapsd 00:08:31

Status
Found IP 10.0.0.9 for NetworkWireless:00:0E:35:38:69:22 via ARP
Found new network "sandyteo" bssid 00:0A:E9:01:D8:F0 WEP Y Ch 6 @ 11.00 mbit
Found new probed network "<no ssid>" bssid 00:0E:35:38:69:22
Saving data files.
Battery: AC 0% 0h0m0s

```

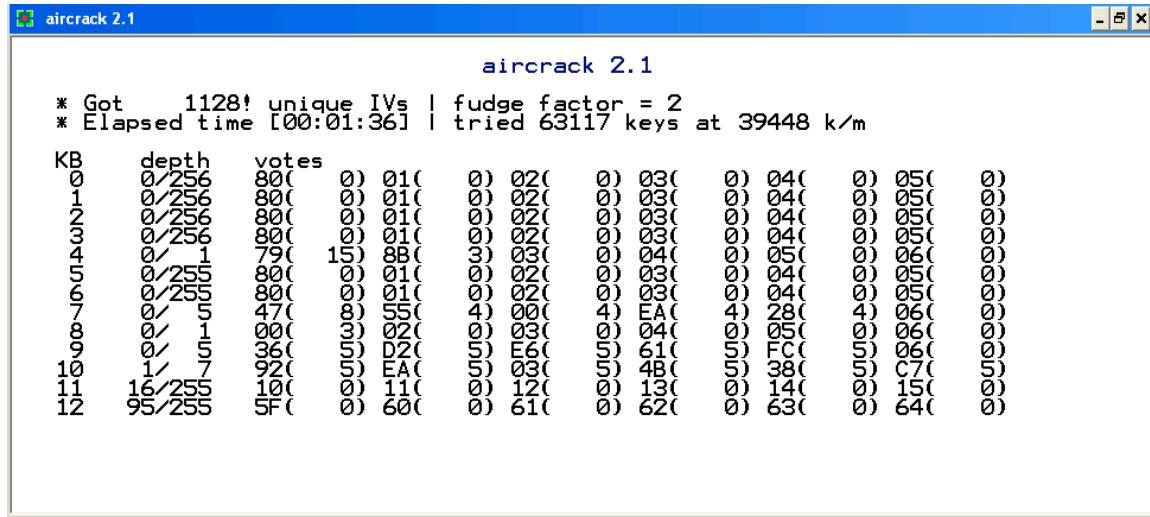
Figure 11 - Kismet

Surveillance

Once the wireless target has been identified, the attacker can proceed to gather information about the network using tools like kismet or airodump (comes with aircrack package at <http://www.cr0.net:8040/code/network/>). Data gathered can be saved into pcap format for subsequent offline analysis.

If the traffic stream is not encrypted, immediately the attacker could look at the traffic stream and identify the network parameters (e.g. MAC address, IP address range, gateway etc) from the traffic.

If the traffic stream is WEP encrypted, there are also WEP crackers which are available for him. In this case, airodump is used to gather all the encrypted packets transmitted and aircrack (see Figure 12) is then used to try to crack the WEP key given enough WEP IVs that are gathered.



```

aircrack 2.1
aircrack 2.1
* Got 1128! unique IVs | fudge factor = 2
* Elapsed time [00:01:36] | tried 63117 keys at 39448 k/m

KB  depth  votes
0   0/256   80( 0) 01( 0) 02( 0) 03( 0) 04( 0) 05( 0)
1   0/256   80( 0) 01( 0) 02( 0) 03( 0) 04( 0) 05( 0)
2   0/256   80( 0) 01( 0) 02( 0) 03( 0) 04( 0) 05( 0)
3   0/256   80( 0) 01( 0) 02( 0) 03( 0) 04( 0) 05( 0)
4   0/1     79( 15) 8B( 3) 03( 0) 04( 0) 05( 0) 06( 0)
5   0/255   80( 0) 01( 0) 02( 0) 03( 0) 04( 0) 05( 0)
6   0/255   80( 0) 01( 0) 02( 0) 03( 0) 04( 0) 05( 0)
7   0/5     47( 8) 55( 4) 00( 4) EA( 4) 28( 4) 06( 0)
8   0/5     00( 3) 02( 0) 03( 0) 04( 0) 05( 0) 06( 0)
9   0/5     36( 5) D2( 5) E6( 5) 61( 5) FC( 5) 06( 0)
10  1/7     92( 5) EA( 5) 03( 5) 4B( 5) 38( 5) C7( 5)
11  16/255  10( 0) 11( 0) 12( 0) 13( 0) 14( 0) 15( 0)
12  95/255  5F( 0) 60( 0) 61( 0) 62( 0) 63( 0) 64( 0)

```

Figure 12 - Aircrack used to crack WEP key

In cases where there isn't sufficient traffic on the network, packet injection tools like WEPWedgie (<http://sourceforge.net/projects/wepwedgie/>) can be employed to insert arbitrary traffic into the WEP encrypted network. This will solicit responses from the network, which can then be collected and send for WEP key cracking. This is made possible because in WEP implementation, as long as one obtains the keystream used for the XOR operation using a single IV value, one can effectively reuse the same IV for all subsequent communications. To obtain a single keystream that corresponds to a particular IV, one has to look for a known plaintext and a corresponding ciphertext in the network. Using that, one can then perform an XOR operation to obtain the keystream used to encrypt the packet. One example where you'll find the plaintext and its corresponding ciphertext is when a wireless client is authenticating to the access point using shared key authentication.

Even when you are unable to get the shared key authentication traffic on the network, tools like chopchop (<http://www.netstumbler.org/showthread.php?t=12489>) which makes use of the access point to help it decrypt 1 WEP encrypted packet at a time without knowing the WEP key is available.

DOS attacks

DOS type attacks at layer 1 as well as layer 2 are easily executed in a wireless network. Emitting a very strong RF interference on the channel in which the wireless network is operating on will increase the noise on that channel and thus causing interference to all wireless networks that are operating at / near that channel.

Layer 2 type DOS attacks come in the form of packet injection, where the attacker will flood wireless clients who are already attached to the wireless networks with disassociate or deauthenticate packets. Example of such a tool is Void11 (<http://www.wi-foo.com/index-3.html>).

Impersonation

Another category of attacks that can be easily executed in a wireless network is the impersonation attack. In such an attack, the attacker changes his MAC address to a MAC address which he found earlier during the surveillance stage. This MAC address would most definitely belong to an authorized client in the network. This is usually done to defeat the MAC filtering capabilities of access points where only a list of authorized MAC addresses are allowed to use the wireless network.

As we have earlier described, even if the wireless network is WEP encrypted, the MAC address of the sending and receiving party is still viewable by a wireless sniffing tool.

Changing the MAC address of your wireless NIC card is made easier by the fact that some client software that comes with the NIC card actually allows the user enter their desired MAC address.

To change the MAC address manually in linux, use the ifconfig command with hw ether as shown below.

```
[root]# ifconfig eth0 hw ether 01:02:03:04:05:06
[root]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 01:02:03:04:05:06
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:4 dropped:0 overruns:0 carrier:4
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:168 (168.0 b)
Interrupt:11 Base address:0xdf00 Memory:df9ff000-df9ff038
```

Figure 13 - MAC spoofing in linux

To change the MAC address manually in Windows, locate the registry settings for your wireless NIC and add a new string call NetworkAddress with the new MAC address information to it. (See figure 14 & http://www.nthelp.com/NT6/change_mac_w2k.htm for details)

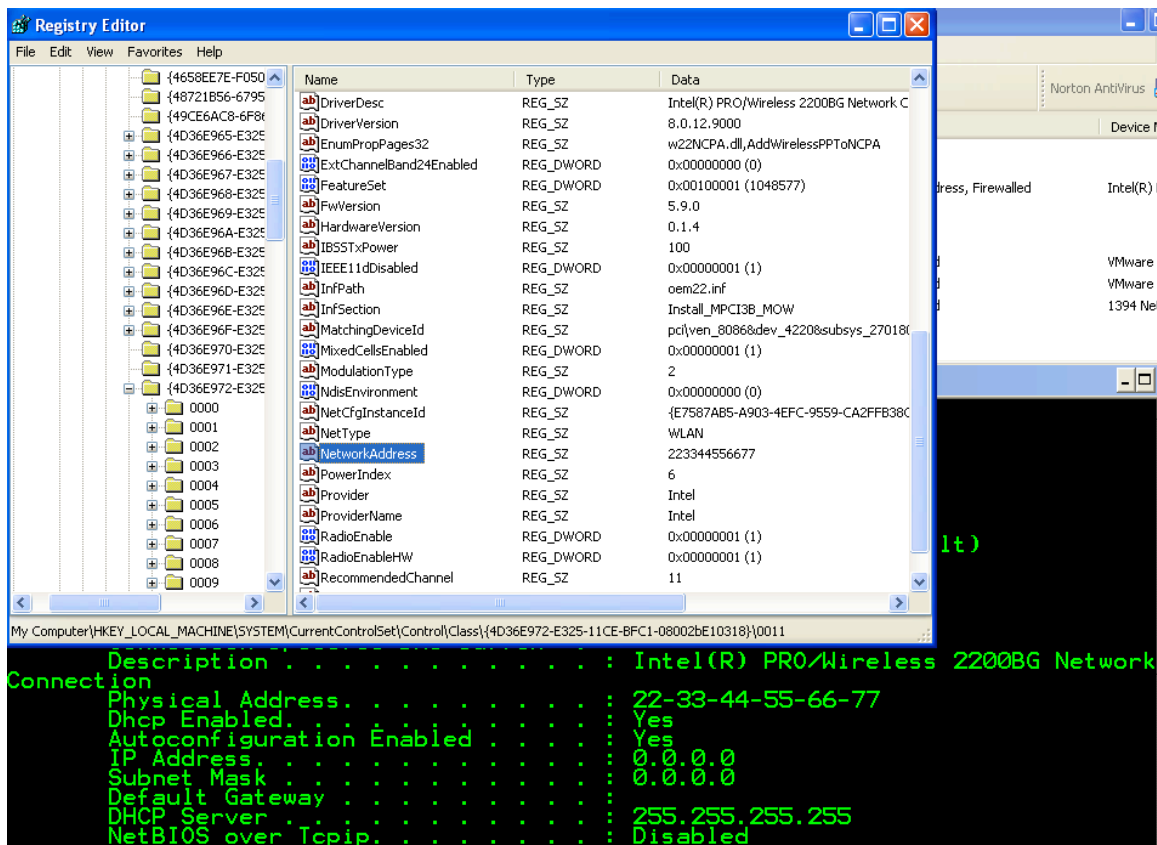


Figure 14 - MAC spoofing in Windows

Man in the middle and Rouge AP

In this type of attack, the attacker attempts to insert himself in the middle of a communication for purposes of intercepting client's data and could potentially modify them before discarding them or sending them out to the real destination.

In order to insert oneself in the middle of the communication, one has to accomplish 2 tasks, first, the legitimate AP serving the client must first be brought down or made "very busy" so as to create a "difficult to connect" scenario for the wireless client, secondly, the attacker must setup an alternate rouge AP with the same credentials as the original for purposes of allowing the client to connect to it.

Either RF interference or layer2 packet flooding as described above in DOS attack section can accomplish the first task.

The second task can be accomplished by setting up a rouge AP that will take over the tasks of the failed AP. Tools like monkey_jack (part of airjack tool found on <http://www.wi-foo.com/index-3.html>) could be employed.



Figure 15 - Man in the middle

Wireless Attacks Detection Techniques

Let's now turn our focus towards wireless attacks detection techniques and look at what should be in place in order to sufficiently detect all of the above mentioned types of attacks.

Access Point Monitoring

This would typically entail the owner of the wireless network having a list of authorized AP equipment with their respective SSID, MAC address, channel information recorded down as a baseline. The monitoring component would then listen to wireless frames (beacons, probe response and authentication / association frames etc) send out by all its AP and compare these information to the pre-recorded information. The monitoring device should listen to all possible channels and record all packets for this technique to be effective.

In the case of a Man-in-the-middle attack, such a component would detect that there is a sudden introduction of an AP on another channel previously not present. Though the SSID, MAC address might be spoofed by the attacker in the process of setting up the rouge AP, the channel in which the genuine AP was operating from has been changed provides an alert on a possible MITM attack.

Wireless Client Monitoring

Unlike APs, it would not be possible to have a list of "allowed" client information baseline without incurring a whole lot of administrative overheads, nevertheless several aspects of the wireless clients can be monitored

Firstly, the owner could keep a “blacklist” of wireless clients that would be checked against all connecting clients, any client within this list trying to access the network would be automatically denied and an alert send off.

Secondly, all wireless clients with an “illegal” MAC address (MAC address ranges which have not been allocated out yet) be automatically denied access and an alert send off.

Thirdly, wireless client that just sends out probe requests or wireless clients that send out special distinguishable data packets after the initial probe request (E.g. Netstumbler sends out a special data packet to solicit AP’s nickname after it receives a probe response from its AP, see <http://www.kismetwireless.net/cgi-bin/ezmlm-cgi?mss:366:eafogdoalgqkiopbclf>) but does not associate / authenticate within a certain period of time can be flagged out as potential network discovery activities.

One more area where monitoring might apply is when WEP traffic is being send/receive, no station should be reusing the same IV over and over again within a very short period of time (we’ve seen this earlier when we describe the use of WepWedgie to generate traffic on your WEP enabled network for purposes of cracking your WEP key)

Lastly, for wireless clients that have been authenticated and associated, the sequence number field within the IEEE 802.11 header can be tracked for sudden changes. Usually when impersonation attacks are underway, the attacker will take on the MAC / IP address of the victim, but it will not be able to continue with the sequence number used previously by the victim, thus by monitoring the sequence number in these client generated packets, potential impersonators could be identified.

General Wireless Traffic Monitoring

Wireless traffic can be monitored for attempts to flood the network using de-authentication, de-association, authentication, association, erroneous authentication (as implemented by Fatajack).

Frequency and Signal-To-Noise Ratio monitoring could help signal an oncoming RF based DOS attack on your wireless network.

Failures in authentication as well as association can also be monitored and reported.

Wireless IDS

We'll look at some examples of open source wireless IDS and evaluate their effectiveness in employing the above mentioned detection techniques.

snort-wireless

Snort-wireless is a wireless intrusion detection system adapted from the snort IDS engine. Hosted at <http://www.snort-wireless.org/>, it adopts the similar syntax when it comes to writing snort-wireless rules as the famous snort IDS.

Replacing the source and destination IP addresses in the normal snort rules with source and destination MAC addresses, one can write snort-wireless rules for detecting wireless traffic like one would detect for IP layer attacks.

As at this point, there are quite a bit of to-do items under the future development. These items are required (as highlighted in the previous section) to effectively be able to address some of the common threats in the wireless world.

WIDZ

This wireless IDS is build by Loud Fat Bloke (Mark Osborne). Hosted at <http://www.loud-fat-bloke.co.uk/tools.html>. The version at the time of this writing is at 1.8. It has the following modules :

- a. Unauthorized AP monitor – responsible for detecting bogus & rouge APs by checking an AP scan result with a baseline file of all authorized APs.
- b. 802.11 Traffic monitor – includes probe / flood monitoring as well as MAC and ESSID blacklist and whitelist.

Though addresses more areas, still there are multiple areas of wireless intrusion detection techniques not addressed at this version.

AirIDS

AirIDS is a wireless intrusion Detection System which is hosted at <http://www.internetcomealive.com/clients/airids/general.php>. It presents a number of interesting aspects to wireless IDS. First of all, like any other IDS, a robust and powerful rules file controls filtering, which is user definable. Also, it is able to forge frames so as to provide not just detection but active defenses against malicious 802.11(b) activities.

Conclusion

To sum it all up, wireless technologies have matured to a stage where it has become very common deployment. Wireless attacks are also evolving as the security standards evolved. Current intrusion detection tools have not matured to a stage where detection is sufficiently reliable.

This paper looks primarily at Layer 1 and Layer 2 type attacks for wireless LAN, but once that is breached, the attacker will then employ traditional types of attack strategies to attack higher layer protocols and applications. Thus defenders ought to look at defense in depth strategies other than just concentrating their efforts in the Layer 1 & 2 type defenses.

As the 802.11i standard is being finalized and rolled out to vendor products, we can expect more attacks to move towards the authentication as well as the encryption technology deployed in these new standards.

In the meantime, enterprises seeking to deploy wireless technologies for whatever reason should stay aware of current standards, software and hardware releases so as to better mitigate the risks brought about by these wireless deployments.

© SANS Institute 2005, Author retains full rights.

References

Andrew A. Vladimirov, Konstantin V. Gavrilenko, and Andrei A. Mikhailovsky. Wi-Foo: Addison Wesley, 2004

Rob Flickenger. Wireless Hacks: O'Reilly, 2003

Chris Hurley, Michael Puchol, et al. WarDriving: Drive, Detect, Defend: A Guide to Wireless Security: Syngress Publishing, 2004

Wright, Joshua. "Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection". 13 Apr. 2005. < <http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf> >

Brenner, Pablo. "A Technical Tutorial on IEEE 802.11 Protocol". 13 Apr 2005. < http://www.sss-mag.com/pdf/802_11tut.pdf >

Loud Fat Blokes. 802.11 Home Page. 13 Apr. 2005 < <http://www.loud-fat-bloke.co.uk/w80211.html> >.

Ossmann, Michael. "WEP: Dead Again, Part 2" Securityfocus. 8 Mar. 2005. 13 Apr. 2005 < <http://www.securityfocus.com/infocus/1824> >.

Wireless LAN Security (802.11) Wardriving & Warchalking. 13 Apr. 2005 < <http://802.11-security.com/security/tools> >

AbsoluteValue Systems, Inc. "WLAN Adapter Chipset Directory". 2 Feb. 2004. 13 Apr 2005. < http://www.linux-wlan.org/docs/wlan_adapters.html.gz >

Kismet. Home Page. 2 Apr. 2005. 13 Apr. 2005. < <http://www.kismetwireless.net> >

Ethereal. Home Page. 11 Mar. 2005. 13 Apr. 2005. < <http://www.ethereal.com> >

CYGWIN. Home Page. 2 Apr. 2005. 13 Apr. 2005. < <http://www.cygwin.com> >

Networking Tools. Aircrack. 13 Apr. 2005. < <http://www.cr0.net:8040/code/network/> >

WebWedgie. Home Page. 13 Apr. 2005. < <http://sourceforge.net/projects/wepwedgie/> >

Korek. "chopchop experimental WEP attacks". Netstumbler Forum. 13 Apr. 2005. < <http://www.netstumbler.org/showthread.php?t=12489> >

Lai, Kyle." Changing the MAC address in W2K and XP". 13 Apr.2005.<
http://www.nthelp.com/NT6/change_mac_w2k.htm >

WI-FOO Tools list. Home Page. 13.Apr.2005.< <http://www.wi-foo.com/index-3.html> >

Craik, Mike."All Your 802.11b Are Belong To Us (Netstumbler
Signature)".Kismet Forum. 13 Apr.2005.< <http://www.kismetwireless.net/cgi-bin/ezmlm-cgi?mss:366:eafojgdoalggkiopbcif> >

Snort-wireless.Home Page.13 Apr.2005.< <http://www.snort-wireless.org/> >

AirIDS. Home Page. 13 Apr.2005.<
<http://www.internetcomealive.com/clients/airids/general.php> >

© SANS Institute 2005, Author retains all rights.



Upcoming SANS Training

[Click Here for a full list of all Upcoming SANS Events by Location](#)

SANS Tampa - Clearwater 2017	Clearwater, FLUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NVUS	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, IE	Sep 11, 2017 - Sep 16, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MDUS	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, DK	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, GB	Sep 25, 2017 - Sep 30, 2017	Live Event
Data Breach Summit & Training	Chicago, ILUS	Sep 25, 2017 - Oct 02, 2017	Live Event
Rocky Mountain Fall 2017	Denver, COUS	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, NL	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Oslo Autumn 2017	Oslo, NO	Oct 02, 2017 - Oct 07, 2017	Live Event
SANS DFIR Prague 2017	Prague, CZ	Oct 02, 2017 - Oct 08, 2017	Live Event
SANS Phoenix-Mesa 2017	Mesa, AZUS	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS October Singapore 2017	Singapore, SG	Oct 09, 2017 - Oct 28, 2017	Live Event
SANS AUD507 (GSNA) @ Canberra 2017	Canberra, AU	Oct 09, 2017 - Oct 14, 2017	Live Event
Secure DevOps Summit & Training	Denver, COUS	Oct 10, 2017 - Oct 17, 2017	Live Event
SANS Tysons Corner Fall 2017	McLean, VAUS	Oct 14, 2017 - Oct 21, 2017	Live Event
SANS Tokyo Autumn 2017	Tokyo, JP	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Brussels Autumn 2017	Brussels, BE	Oct 16, 2017 - Oct 21, 2017	Live Event
SANS Berlin 2017	Berlin, DE	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS San Diego 2017	San Diego, CAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, AE	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FLUS	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Amsterdam 2017	Amsterdam, NL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Milan November 2017	Milan, IT	Nov 06, 2017 - Nov 11, 2017	Live Event
Pen Test Hackfest Summit & Training 2017	Bethesda, MDUS	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Paris November 2017	Paris, FR	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS Sydney 2017	Sydney, AU	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS San Francisco Fall 2017	OnlineCAUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced