You've Been Hacked: Analyzing Wireless Security in Columbia, MO

Matthew Chittum, Clayton Harper, John Mixon, Johnathan Walton

Advisors: Dr. Wenjun Zeng, and Dr. Haibin Lu

University of Missouri – Columbia

The current state of wireless security in most areas can be estimated based on trends and collected data, but the complete picture is often unknown. Without collecting the information on most wireless access points in a given area, we cannot compare different areas based on their security, get an accurate view of the areas as a whole, or relate wireless security with other factors in a given region. We have performed a wireless security audit of Columbia, MO. Information was collected on thousands of access points throughout the city in hopes of understanding a large area's use of wireless networking and its possible security flaws. The flaws of WEP encryption allow a supposedly secure computer to be breached and then compromised. We have demonstrated how simple it is to bypass WEP encryption using easily accessible software available on the Internet. We have determined that although Columbia's overall level of security is better than the national average, it still leaves much room for improvement.

1. Introduction

The Internet has become a place where millions of people have replaced physical services for virtual ones. Online shopping, banking, stock trading, and gambling have thrown people's private financial information into the web. Since the advent of commerce there have been people trying to take advantage of others to dishonorably make profit. With people's credit card numbers, bank information, and other private data flowing through networks, Internet criminals try their best to steal this data.

Now that mobile computing is extremely common, much of this private data is being sent over the air through the use of wireless network devices. As described in the introduction, packets that flow from a computer to an access point, potentially containing private data, can be grabbed in transit from by neighboring computers. Though the packets may have a layer of encryption protecting easy access to the data, with enough packets a hacker may be able to compromise the security and retrieve the data. With the correct amount of security enabled by a user, these potential hazards can be possibly evaded. The real problem has now been revealed. Yet another security hazard of wireless networking is an unsecured access point. As described in the introduction, a wireless access point is connected to a home network and provides access to that network for mobile devices. A mobile device could therefore possibly compromise other computers on that network, obtaining information from wired computers.

According to AP tracking site Wigle.net, upwards of 40% of Wireless APs have no security enabled. This means that those networks broadcasting an insecure signal are prone to attack.

2. Methodology

To evaluate the security of Columbia, we used a popular AP detection technique called wardriving. The basic principles of wardriving include multiple AP detection while logging the GPS coordinates of those APs. The best method is to drive around detecting APs and logging the information about each AP. Any transportation will do, as people can warwalk, warbike, or even warfly. Wardriving requires certain tools. Obviously, some kind of transportation is necessary so that a decent area can be logged. Secondly, a mobile computer with a wireless card is required to collect the packets sent by the AP. A software package eases the process of finding, collecting, and storing the information about each AP detected. The software package used in this research was Netstumbler (www.netstumbler.org). Netstumbler causes the wireless card in the computer to send out packets that request information from local detectable APs. The APs in range then send out packets containing useful information such as SSID, MAC address, and type of 802.11 standard, among others. The essential aspect of wardriving is logging the location of the APs detected. This can be done by connecting a GPS unit to the computer that is detecting the APs. The GPS must be able to send its latitude and longitude to the computer. The GPS that was used in this research (Garmin eTrex) sent the current position coordinates to the computer every two seconds. Netstumbler is compatible with any GPS connected to the computer that is NMEA compliant. While tracking APs Netstumbler would log the GPS coordinates associated with that AP's location. All that is left to do is drive around a given area and the computer will log all APs detected during the drive.

There are ethical considerations for wardriving, as you are technically obtaining information about a person's private wireless network. Wardriving is under debate, but the current law says it is legal. However, the law says that using someone else's wireless Internet is illegal. Netstumbler simply does what the wireless card is designed to do, it only logs the information retrieved. At no point does Netstumbler use those networks for anything. The information received during wardriving does detail whether a specific AP is secure or not, this information could be used for illegal activity, but the simple request for that information does not constitute breaking the law.

A major concern regarding the state of wireless security is the number of networks that are protected by WEP security. WEP can be broken fairly easily as discussed in the details section.

3. Details

The details of wardriving are not too complex, but having the correct approach is essential. Before we discuss the results and what we have learned from this experience, it is important to detail the road taken. Wardriving relies on packet sniffing, GPS synchronization, and wireless network security protocol to effectively measure an area's security. In the following, each will be investigated.

The idea of packet sniffing can be determined from its name. The essential principle is that a wireless network adapter that can be turned into "passive" or "promiscuous" mode is able to retrieve all packets that come within range of the receiver, regardless of the intended recipient. The computer with this passive wireless adapter is now able to store these thousands of packets it receives and analyze them for an intended purpose, most often in an attempt to crack whatever security the transmitting AP has, or to look into the packets for information such as credit card numbers or ATM codes. Packet sniffing is most essential to cracking WEP security. The packets contain information pertaining to unique IVs essential to the WEP process, and the collection of thousands of these allows a computer to break WEP security. This process is detailed later in this section.

A packet sniffer must obviously have packets to sniff, and there is not always network traffic on an AP. To successfully pick up information about all detectable APs within range, we must have a way to trigger them to send packets if they are not already. A program such as Netstumbler sends probes out and local APs respond by sending their information. There isn't any incriminating information in these packets, though a

computer could be set up to continuously send these requests until enough packets are retrieved for an attack.

GPS (Global Positioning System) is a technology launched in the 1980s for military use but has since found widespread civilian adoption. There are 24 satellites in orbit around earth transmitting data via low power radio waves. Many pursuits such as cartography, navigation, and communication now rely on GPS technology. GPS provides 2-Dimensional positioning and 3-Dimensional positioning, along with speed, bearing, and various distances. The essential information for wardriving is of course position. While the computer is sniffing packets, the program that records the SSID and MAC addresses obtained associates with them a GPS coordinate, providing a position fix to the related access point. GPS proved invaluable to this research; using it along with Google Maps provided an overlay of AP locations and information on top of a satellite view of Columbia, MO.

The following details the steps we took to wardrive:

Tools Required:

Computer running Windows XP
Netstumbler v 0.4.0
Garmin Etrex and serial cable
Dell Wireless 1370 WLAN Mini-PCI card
Earthstumbler
Google Earth
Car

Steps:

- Download and install necessary software
  - Download Netstumbler v 0.4.0 from http://www.netstumbler.com/downloads/ and install.
  - Download Earthstumbler from http://mboffin.com/earthstumbler/
  - Download Google Earth from http://earth.google.com/ and install.

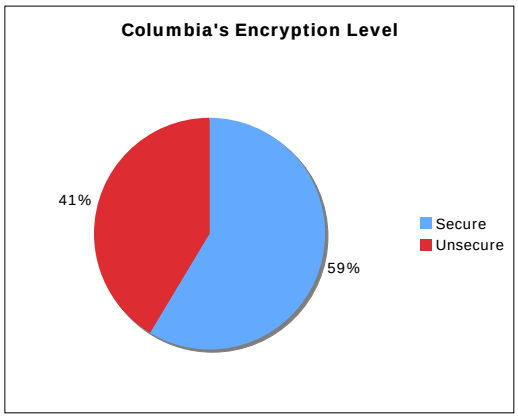
- Setup Netstumbler and Garmin ETrex.
  - **Setup Garmin Etrex:** On the Garmin Etrex hit the "Page" button multiple times until you get to the "Setup" screen. Use the arrow keys and go to "Interface" and press enter. Change the I/O format to NMEA Out and the baud rate to 4800.
  - **Setup Netstumbler to accept GPS input:** In Netstumbler go to View -> Options -> GPS (tab) and change protocol to NMEA 0183 and the baud rate to 4800.
  - **Setup serial port:** In device manager change the baud rate of the serial port you are using to 4800.
  - Make sure the GPS is tracking the satellites and make sure netstumbler is collecting the GPS data. If it is not working go to the netstumbler forums and ask for help.
  - Once the GPS is setup and working you can drive around and collect data. Check it every few minutes to make sure it is collecting data.

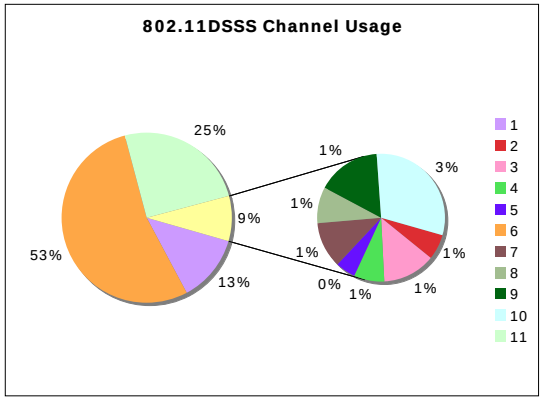
- View data in Google Earth:
  - **Export a summary file:** Once you've collected enough data export a summary file which is needed with Earthstumbler. In Netstumbler go to File->Export->Summary.
  - **Earthstumbler:** Open the summary file you exported above in Earthstumbler. Save the file.
  - **Google Earth:** Open the file Earthstumbler created. You should be able to see the Access Points you captured and where you captured them.

Columbia, MO gives an interesting picture of wireless security. Through a series of wardrives, 5563 unique APs were collected. An analysis of this information paints a portrait of better than average security, though far from perfect.

There are many points of interest to look at:

- 59% of discovered APs were secure, nearly 20% higher than the national average [1].





- 88.1% of APs had 802.11g capability.
- Of the 5,563 APs we surveyed in Columbia, nearly 30% had a default SSID, 12% higher than the national average [1].
- It is possible to crack WEP security in as little as 10 minutes in a heavy traffic area; campus wireless TigerNET uses WEP security.
- Channels 1, 6, and 11 comprise 91% of 802.11DSS channels used.

Wireless Encryption Protocol is a protocol used to secure IEEE 802.11 wireless networks. The protocol is described briefly below; it is discussed in much more detail in 802.11 standard [4].

The WEP protocol begins with the message (M) to be transmitted. A cyclic redundancy check (CRC) is run on the message (M) and the result of the CRC is then appended to the original message (M) to the form the plaintext (P). The plaintext (P) is then encrypted using the RC4 algorithm, which is a function of both the Initialization Vector (IV) and the secret key (k). The IV is always 24 bits and is usually linearly incremented (incremented by one for each data packet being sent). The secret key is shared only between the transmitter and receiver and is either 40 bits or 104 bits. The RC4 algorithm (a stream cipher algorithm) generates a keystream (B) of pseudorandom bytes using both the IV and the keystream. The keystream is either 64 bits or 128 bits depending on the size of the secret key that was used. The keystream is then XORed with the plaintext to form the ciphertext (C). The IV is appended to the ciphertext and then both the IV and ciphertext are transmitted. It is important to note that the IV is sent "as is" and is not encrypted.

An overview of the WEP encryption process that uses equations [5]:

$$P = M + CRC$$
$$B = RC4(IV, k)$$
$$C = P \oplus B = P \oplus RC4(IV, k)$$

Decryption of a data packet with WEP encryption is uncomplicated because it consists of reversing the encryption process that is done by simply regenerating the keystream and XORing it with the ciphertext. An overview of the decryption process using equations [5]:

$$P' = C \oplus RC4(IV, k)$$
$$= (P \oplus RC4(IV, k)) \oplus RC4(IV, k)$$
$$= P$$

We will briefly look at various attacks that can successfully crack a wireless network encrypted with WEP [1].

1. **Brute Force** - The brute force method is the simplest method because it tries all possible key combinations until the correct key is found. Because of the length of keys WEP uses it takes an extremely long time for this method to successfully find the correct key and with the introduction of the 104-bit key the length of time is even more pronounced. Therefore while this method does work it is very inefficient and is not considered practical.
2. **Keystream reuse** – If a keystream is known then it is possible to recover the data that was encrypted using that keystream. However, in order to uncover the keystream the plaintext must be known. The RC4 algorithm that WEP uses has a notable problem in that encrypting two different messages using the same IV and secret key can reveal important information about both messages. Since only 2^{24} (16 million) IVs exist (and even less if weak IVs are excluded) IVs can be repeated within the matter of hours. If ciphertexts from both messages with duplicate IVs are XORed together then the result is both of the original plaintexts XORed together. The individual plaintexts can then easily be found by searching for two English phrases that when XORed together form the two plaintexts XORed together. An overview in equation form is given below [5]:

$$C_1 = P_1 \oplus RC4(IV, k)$$
$$C_2 = P_2 \oplus RC4(IV, k)$$
$$C_1 \oplus C_2 = (P_1 \oplus RC4(IV, k)) \oplus P_2 \oplus RC4(IV, k))$$
$$= P_1 \oplus P_2$$

3. **Weak IV** – The secret key can be computed by capturing many packets some of which use "weak" IVs. One weak IV can reveal a correct key byte 5% of the time. With a large number of IVs the most probable key can be guessed [2, 6].

To effectively crack a WEP secured network, it is important to collect as many packets as possible. The most common cracking method is a Weak IV crack. This procedure relies on many packets. Not every AP is in a heavily trafficked area; therefore it may be necessary to cause the AP to generate some packets on your own. There are a few ways to do this; one way is an Authentication Flood. An Authentication Flood involves sending many requests for service to an AP by a computer that is generating random but believable MAC addresses. This service flood has the AP responding to each request with more packets, thus generating traffic for the sniffer to collect. Another similar attack is Packet Re-injection. Packet Re-injection requires capturing a packet of guessable size, which is essentially a request packet. This packet is then "re-injected" to the AP multiple times, thus generating network traffic.

We will look briefly at the times it takes to crack WEP. Since these times have been documented extensively in other similar papers and we didn't have the adequate means (in terms of hardware and software) to perform attacks attempting to crack WEP we will simply reference several other papers that detail the times associated with certain known WEP cracking methodologies and allow you to explore these yourself.

“We demonstrate an active attack on the WEP protocol that is able to recover a 104-bit WEP key using less that 40,000 frames with a success probability of 50%. In order to succeed in 95% of all cases 85,000 packets are needed [3].”

“With 40 bit keys, the median number of packets required to crack the key is one million. With two million packets, 80% of the 40-bit key could be obtained.” (Note: The graph below is also associated with preceding text) [1].

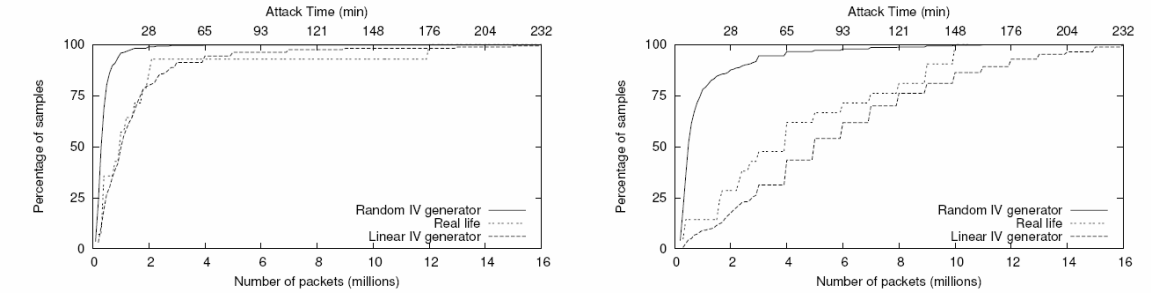


Figure 8. Cumulative distribution of packets required for cracking 40 bit (left plot) and 104 bit keys.

4. Conclusions

Throughout this paper we have analyzed the security of Columbia, MO. The statistics obtained from wardriving show that Columbia has better than average security, but is far too reliant on WEP security. More people should be made aware of WEP's shortcomings and offered alternatives such as WPA. Future work in the realm of analyzing a community's security is to actually talk to residents and see if they are aware of what kind of security they use and what they know about wireless security in general. With regards to wireless security, work should be done continuously in the pursuit of more secure standards and attempts to find flaws in those and other existing standards. One of the potential problems with wireless security is that there will always be some way to break the protocol and gain access to the system.

References:

[1] A. Bittau, M. Handley, and J. Lackey. The Final Nail in WEP's Coffin. http://tapir.cs.ucl.ac.uk/bittau-wep.pdf

[2] D. Wagner. Weak Keys in RC4, 1995. www.cs.berkeley.edu/~daw/my-posts/my-rc4-weak-keys

[3] E. Tews, R. Weinmann, A. Pyshkin. Breaking 104 bit WEP in less than 60 seconds. http://eprint.iacr.org/2007/120.pdf

[4] L.M.S.C of the IEEE Computer Soceity. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11, 1999 Edition.

[5] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf

[6] S. Fluhrer, I. Mantin, A, Shamir. Weakness in the Key Scheduling Algorithm of RC4. www.cs.umd.edu/~waa/class-pubs/rc4_ksaproc.ps

[7] Wigle.net Statistics as of July 31, 2007 http://www.wigle.net/gps/gps/main/stats/