University of Windsor

E-mail Capturing & E-mail Encryption

Mohit Sud
10/21/2009

E-mail Capturing & E-Mail Encryption

Table of Contents

Executive Summary

This project summarizes the lack of privacy that exists with one of the most largely used forms of communication, e-mail. Once an e-mail message is sent from its sender to its recipient, neither party involved have any control on who may see it, or on what server it may be stored on. The focus of this project is to demonstrate the vulnerability and insecurity that exists when sending an e-mail message, and then to provide a method of ensuring the privacy of an e-mail through the use of cryptography.

This document contains a brief tutorial on how an intruder can easily obtain access to an otherwise secure wireless network using the vulnerabilities that exist in the WPA wireless encryption scheme. The WPA encryption scheme is widely used by both home and commercial networks as a method of keeping a wireless local area network private and secure. This project exploits a vulnerability included in the WPA encryption to obtain the passphrase required to successfully gain access to that network.

The tutorial continues to explain how through the use of packet sniffing someone may eavesdrop on all of the incoming and outgoing data on the local area network to which it may be connected to. Packet sniffing is a term used to explain the concept of collecting and analyzing network traffic. This traffic may include e-mail messages and web site passwords. An example is given where an e-mail message is sent from a victimized computer, and how a second computer (also connected to the same network) is able to collect the outgoing packets, reassemble them, and read the contents of the full e-mail message.

Through the decryption of the WPA passphrase and the capturing and analyzing of packets via the technique of packet sniffing, it is clearly evident that e-mail messages are insecure and vulnerable to anyone with the knowledge and motivation to read them. Thus, a method is provided that an individual may use to encrypt the contents of an e-mail message at the sending end, and have it decrypted at the recipient. This technique is accomplished through the use of PGP. PGP refers to an encryption algorithm that uses the public key/private key approach to disguise the contents of an e-mail message as a random jumble of characters known as ciphertext. It is not until the recipient enters there secret passphrase that the e-mail message is converted back into plain text.

Through the examples demonstrated on how easily a trespasser may read another persons e-mail message, it is necessary to consider the security and privacy behind the messages we exchange. It leads us to the conclusion that it is a worthwhile endeavour to incorporate the tactics of e-mail encryption in order to ensure our data remains private.

Introduction

Overview

Electronic mail (e-mail) messages are one of the most common forms of communication in today's world. It has made its way to be one of the most preferred methods of communication by many people. Its success has been attributed to many factors, including; low cost, speed, reliability, and ease of use; but what about security and privacy? As an e-mail message is sent from sender to recipient, it will travel vast distances, crossing through multiple networks (both secure and insecure), and making copies of itself on servers all across the internet. Since e-mail messages are traditionally written in plaintext, anyone with access to those servers or gathering outgoing/incoming packets can easily read your e-mail. This results in the conclusion that e-mail messages are highly insecure.

Objective

The task of ensuring that no copies are made of an outbound e-mail to any server on the network is an enormous and unfeasible approach. Once an e-mail message is sent, we have no control over what happens to it. As well, the task of ensuring packets are not being snooped on is a large responsibility and may not always be guaranteed. Thus, a different approach must be undertaken to ensure the privacy behind e-mail messages. The objective of this project is to demonstrate how vulnerable and insecure an individuals e-mail messages are, and the importance behind using e-mail encryption to ensure privacy. Real examples will be performed where we will simulate how an intruder may gain unlawful access to a WPA encrypted network, and furthermore how that intruder can gather network data and read an individual's e-mail messages without that persons consent or acknowledgement.

Project Scope

Essentially, we will demonstrate how a hacker may gain unauthorized access to an otherwise secured WPA network, and furthermore how that hacker may then perform various packet sniffing techniques to read a network users e-mail and web traffic. A solution of how to encrypt e-mail messages will also be demonstrated to ensure ones privacy is ensured.

Packet Sniffing

What is Packet Sniffing?

As data is transmitted over the internet, it is divided into small chunks called packets. A collection of these packets will create the final information that will be seen by the receiver (website or e-mail). A packet sniffer allows an individual to gather these packets (intercept or copy them) and see any bit of information entering or leaving a computer.

A packet sniffer may be considered as a wire-tap device. It is a tool that can eavesdrop on the network traffic.

How Packet Sniffing Works

Packet sniffers are designed to make it as simply as possible on the user. It captures 'binary' data that is passing through the network that it is associated with. Most, if not all packet sniffers 'decode' the data that it gathers into a human readable form. Packet sniffers may also contain 'protocol analysis' which may further break down the packet information and give complex details about what is in it. This data may include a password for a specific web service.

In order for packet sniffing to work successfully, you have to be connected on a LAN that uses a hub and not a switch. Suppose 4 computers (A, B, C & D) are all connected through a hub. Computer A wants to send something to computer D, so it starts by sending that information to the hub. The hub does not know where D is, so the hub will re-transmit what A sent to all other computers on the network. Computer C and B should expectedly ignore the data since the packet says its for computer D. The security issue with this scenario is that all computers on the network are receiving all packets, regardless if it was meant for them or not. A packet sniffer can listen in on this data even though it may not be meant for that specific computer.

Simply because packet sniffers require a hub to work effectively, this does not mean that computers behind a switch are secure. Although a switch has the intelligence to know where the information is going and does not forward it to all computers on the network, there are still methods to force a switch to behave as though it were a hub. Some of these methods include flooding the switch with ARP requests or tracking the switch into redirecting traffic to the sniffer system.

It is important to note that the packet sniffer must be on the same network (LAN) on which the data is travelling to. In short the "probing" device that "captures" the data has to be on the same wire. The data can then be relayed to a decoding computer on a different network. Although you may feel your data is secure since you may trust everyone on your network, as we discuss later in this report, it is very easy for someone to gain unlawful access into your network. There are several vulnerabilities in WEP and WPA that may be exploited to give someone access to your network. And once they can connect to your network, they can begin sniffing your packets and seeing exactly what you are doing online.

PGP

What is PGP?

PGP (short for Pretty Good Privacy) is an encryption program that was originally developed by Phil Zimmermann in 1991. PGP uses the Conventional and Public Key Cryptography technique in its application. The traditional key sizes for PGP can be from 512 bits, up to 4096 bits.

GP is an application program that adds a layer of privacy to an otherwise vulnerable electronic mail message. GP accomplishes this feat by utilizing the methodology incorporated by PGP. It adds privacy to e-mail messages by encrypting your mail so that nobody but the intended person can read it. Once the message has been encrypted, it appears as a meaningless jumble of random characters. It's true message can only be read once it has been deciphered using a private key.

Another use of PGP is that it may be used to apply a digital signature to a message without encrypting it. This approach is best used when the sender does not want to hide what they are trying to say, but rather to allow others to verify that the message actually came from them. Once the digital signature has been applied to a message, it is impossible for anyone to modify the message or the signature without the modification being detected by PGP.

Supported Platforms

PGP is an open source application that has been re-written in multiple languages and is supported across many different platforms. A few supported platforms include; Windows, Unix, MS-DOS, OS/2, Macintosh, and Atari.

How PGP Works

Before we can fully understand how PGP works, we must first understand what encryption, decryption, and cryptography is all about.

Encryption & Decryption – The terms plaintext and cleartext refers to data that can be easily read and understood without any special measures being taken. The process of disguising plaintext (or cleartext) in such a manner that its substance is hidden is known as encryption. The encryption of plaintext results in an unreadable jumble of letters and numbers called ciphertext. The purpose of encryption is to ensure that information is hidden from anyone for whom it is not intended for. The technique of reverting ciphertext to its original plaintext form is known as decryption.

Cryptography – The use of mathematics to encrypt and decrypt data is what is known as Cryptography. Cryptography enables the storing and/or transmitting of sensitive information across an insecure media so that it cannot be read and interpreted by anyone except the intended recipient.

Conventional Cryptography – For secure and private communication to take place between a sender and recipient, they both must agree upon a single key and keep it secret between themselves. However, if the key is intercepted, messages can then be read, modified, and forged. The advantages of conventional cryptography is that it is a fast method of encryption.

Public Key Cryptography – In order to resolve the single key vulnerability issues with conventional cryptography, the Public Key Cryptography scheme uses a pair of keys for encryption. A public key is used to encrypt data, while a private (secret) key is used for decryption. The public key is published and can be used by anyone who wishes to send you a message. The private key is kept as a secret and is used to decrypt the message that was encrypted using the public key

PGP can be considered a hybrid cryptosystem. It combines some of the best features of both conventional and public key cryptography. When PGP is first used to encrypt plaintext, PGP first compresses the plaintext. Data compression saves disk space, transmission time, and also strengthens cryptographic security. PGP then creates a random session key. This key is a random number that is generated based upon keystrokes and mouse movements. The key is used in conjunction with a fast encryption algorithm to encrypt the plaintext, resulting in ciphertext. After the data has been encrypted, the session key is then encrypted to the recipient's public key. This public key-encrypted session key is transmitted along with the ciphertext to the recipient.

The PGP decryption works similarly but in reverse. The recipient uses there private key to recover the session key, which PGP then uses to decrypt the conventionally-encrypted ciphertext.

The combination of the two encryption schemes combines the convenience of a public key encryption with the speed of conventional encryption. Conventional cryptography can be up to 1,000 times faster than public key cryptography. However, public key cryptography provides a solution to key distribution issues. Combining these two techniques together provides us with a strong encryption & decryption technique that allows for greater performance without sacrificing security.

WPA

What is WPA?

WPA (short for Wi-Fi Protected Access) was developed by the Wi-Fi Alliance to resolve the security weaknesses associated with WEP security. WPA is an interim standard that was created to provide more security encryption while the 802.11i standard was being approved. It is a security software upgrade from WEP, and can be implemented with no hardware changes to a WEP enabled router.

In WPA, each information packet is encrypted with a different code, or key. Since the key is constantly changing, WPA is very secure. The encryption key is automatically generated from a string of characters known as the pass phrase or from the use of a Pre-Shared Key (PSK). A Pre-Shared Key (known as WPA Personal) is generated using an encryption algorithm known as Temporal Key Integrity Protocol (TKIP). A PSK can be any combination of letters and numbers ranging from 8 to 63 characters.

All clients wishing to connect to a WPA encrypted network must use the same pass phrase or Pre-Shared Key (PSK).

How WPA Works

WPA was designed to resolve the weaknesses associated with WEP encryption. Among these weaknesses, WPA resolves the weak WEP headers issue (known as initialization vectors), and provides a way of ensuring the integrity of packets passed through the Message Integrity Check (MIC). This feat is accomplished through the use of TKIP (short for Temporal Key Integrity Protocol). TKIP is used to generate a Pre Shared Key for each individual packet being transferred.

WPA can be used in both enterprise and consumer environments. The consumer implementation of WPA is known as WPA-PSK. WPA-PSK is a special mode of WPA that provides the same level of encryption without the use of an enterprise authentication server.

WPA-PSK is a strong encryption scheme where encryption keys are automatically changed (known as rekeying). Also, WPA-PSK forces re-authentication between devices after a specified period of time or after a specified number of packets have been transmitted (known as the rekey interval). These techniques allow WPA to be far superior compared to the WEP encryption for two primary reasons. The process used to generate the encryption key is very rigorous and the rekeying is done very rapidly. This stops even the most skilled hackers from gathering enough data to break the encryption.

WPA-PSK is an easy to use method of securing a network. WPA uses a passphrase or Pre-Shared key that must be entered in both the wireless access point (or router) and all WPA clients. The passphrase or Pre- must be between 8 and 63 characters, and may include numbers, letters, special characters and spaces. The WPA Pre-Shared key should be a random sequence of keyboard characters and at least 20 characters in length. It may also be at least 24 characters in length if using hexadecimal digits The more random your WPA Pre-Shared key, the safer it is to use.

The Temporal Key Integrity Protocol (TKIP) begins to handle the encryption and automatic rekeying after the initial shared secret is entered in your wireless devices.

Packet Sniffing (E-Mail Capturing)

Overview

The use of packet sniffing may allow an unwarranted individual the opportunity to read otherwise confidential information from a network user. Outlined in our trial is how one may collect network traffic in the form of packets and analyze these packets to read its underlying content. This information may include passwords and e-mail messages.

Test Environment

Hardware

In order to snoop through network traffic, two computers are required and will be connected via a hub. One computer will be the victim and the other is the eavesdropper. The victim will be sending out various web traffic including e-mail messages and http traffic. The eavesdropping computer will be running packet sniffing software to gather the packets being sent from the victim computer. The eavesdropping computer will analyze the gathered packets and view the underlying content contained. This essentially will allow the eavesdropping computer to read the various emails and web sites of the victim computer.

Product Type	Model	Specifications
Laptop Computer	Toshiba Portege R500	Microsoft Windows XP SP3 Intel Centrino Duo 1.2Ghz.

		1gb. DDR2 RAM 160gb hard drive Intel 3945ABG Pro/Wireless 802.11g
Desktop Computer	Custom Build	BackTrack 3 Live CD Intel Core 2 Duo 2.2Ghz. 2gb. DDR2 RAM 250gb hard drive D-Link 510N 802.11g Wireless Card with Prism
Hub	Linksys 4 Port Hub	10/100 MB
High Speed Internet	Cogeco	10mbps Download Speed 3 IP Addresses

Software

All of the software required for WPA decryption is included on the Backtrack 4 Linux Live CD. Amongst these applications include Aircrack. Aircrack is a set of tools used in WEP/WPA decryption. Also, the madwifi drivers are required to enable 'injection mode' on the wireless card being used.

Software Title	Description	Availability
Microsoft Windows XP	The Operating system	
Wireshark	The packet sniffing & analyzing tool.	http://www.wireshark.org/
Mozilla Thunderbird	The e-mail client sending out an e-mail.	http://mozzila.com/

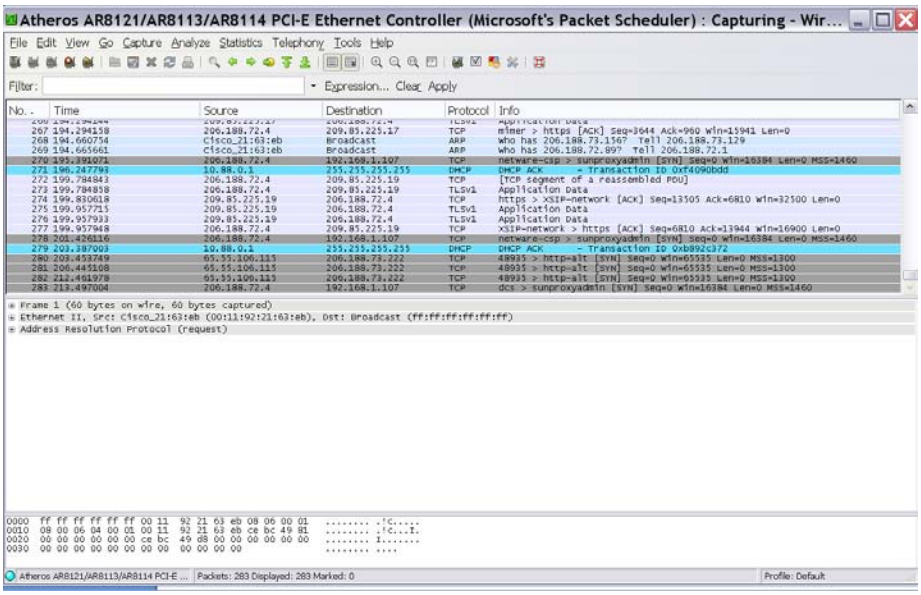
Installation & Configuration

1. Install Mozilla Thunderbird on the victim computer that is running the Windows XP Operating System.
2. Configure Mozilla Thunderbird with the information for our e-mail account.
3. Install Wireshark on the eavesdropping computer.
4. Connect all computers directly to the HUB.

Trial

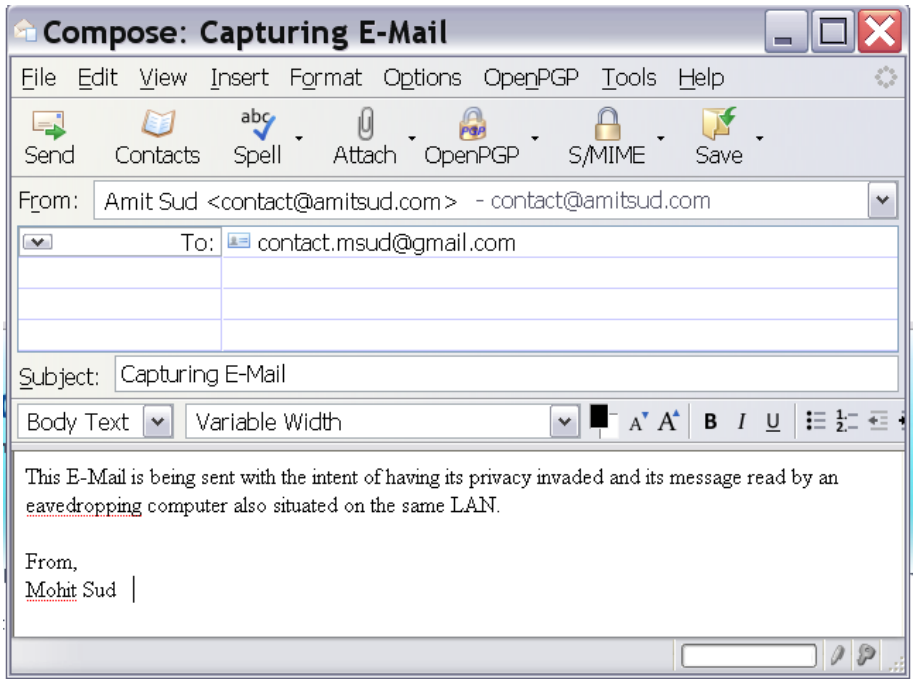
The general procedure to begin packet sniffing is as follows:

1. Begin capturing packets on the network by selecting Capturing -> Start.



Wireshark automatically colour codes the different packet types.

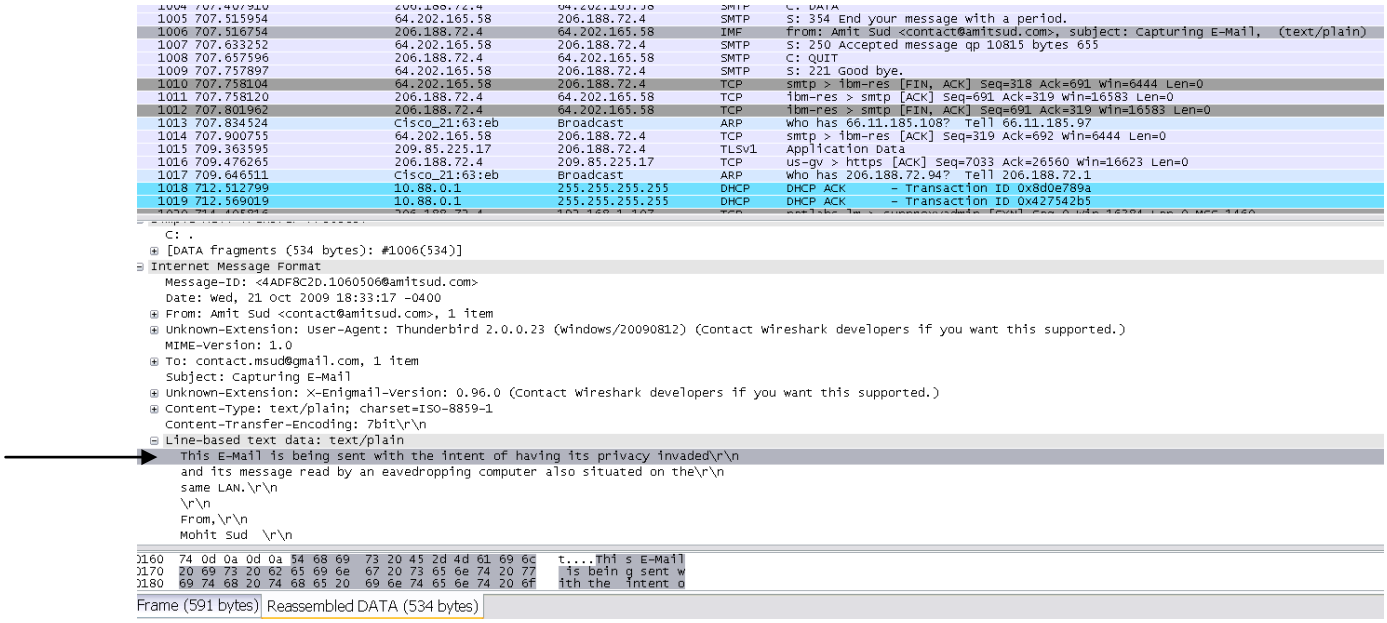
2. Have the victimized computer send out any plain unencrypted e-mail message through Mozilla Firefox.



3. Notice how Wireshark immediately captures the SMTP packets being sent.

602	400.336230	72.167.82.80	206.188.72.4	SMTP	S: 220 p3plsmtpa01-02.prod.phx3.secureserver.net ESMTP
603	400.336901	206.188.72.4	72.167.82.80	SMTP	C: EHLO [206.188.72.4]
604	400.432690	72.167.82.80	206.188.72.4	TCP	smtp > cadsi-lm [ACK] Seq=54 Ack=22 Win=5840 Len=0
605	400.432888	72.167.82.80	206.188.72.4	SMTP	S: 250-p3plsmtpa01-02.prod.phx3.secureserver.net
606	400.585382	206.188.72.4	72.167.82.80	TCP	cadsi-lm > smtp [ACK] Seq=22 Ack=101 Win=16800 Len=0
607	400.682324	72.167.82.80	206.188.72.4	SMTP	S: 250-AUTH LOGIN PLAIN | 250-8BITMIME | 250 PIPELINING
608	400.682636	206.188.72.4	72.167.82.80	SMTP	C: AUTH PLAIN AGNvbnRhY3RAYW1pdHN1ZC5jb20AZnVqaWZpbG0=
609	400.788791	72.167.82.80	206.188.72.4	SMTP	S: 235 Authentication succeeded.
610	400.788998	206.188.72.4	72.167.82.80	SMTP	C: MAIL FROM:<contact@amitsud.com>
611	400.888555	72.167.82.80	206.188.72.4	SMTP	S: 250 Sender accepted.
612	400.888749	206.188.72.4	72.167.82.80	SMTP	C: RCPT TO:<contact.msud@gmail.com>
613	400.984120	72.167.82.80	206.188.72.4	SMTP	S: 250 Recipient accepted.
614	400.984323	206.188.72.4	72.167.82.80	SMTP	C: DATA
615	401.095377	72.167.82.80	206.188.72.4	SMTP	S: 354 End your message with a period.

4. Analyze the SMTP packets to obtain the e-mail message contents.



If you will notice at the bottom in the packet analysis section, the e-mail message is visible in plaintext.

Results

After enabling Wireshark to eavesdrop on the network traffic, collect packets, and analyze those packets, it is clearly evident that e-mail messages are insecure and can easily be read by anyone willing to put forth the effort in doing so. The screenshots above clearly depict the SMTP mail packets going out, and an analysis of those packets reveal the contents of those messages. All the necessary information; sender, recipient, subject, and body are all present. Thus, this leads us to the conclusion that our e-mail messages are insecure and that there is a demand for a method of securing the privacy of our e-mail messages.

WPA Decryption Technique

Overview

There are various vulnerabilities in the WPA encryption technique that can be exploited to allow an unauthorized user access onto a network. In this section, we will explore what these vulnerabilities are and exploit them to gain access to a WPA encrypted network. WPA traditionally uses a passphrase as a means authenticate a user with its network. Using our test environment, we will attempt to retrieve what that passphrase is for a WPA encrypted network by using various techniques including; sniffing, handshake collecting, and brute force.

Test Environment

Hardware

In order to decrypt a WPA networks passphrase, the following tools are required. Firstly, a router that is capable of supporting the WPA encryption must be included. In addition, at least 1 client must be connected to that router. And lastly, a PC is needed to attempt to decrypt the passphrase for that network.

Product Type	Model	Specifications
Laptop Computer	Toshiba Portege R500	Microsoft Windows XP SP3 Intel Centrino Duo 1.2Ghz. 1gb. DDR2 RAM 160gb hard drive Intel 3945ABG Pro/Wireless 802.11g
Desktop Computer	Custom Build	BackTrack 3 Live CD Intel Core 2 Duo 2.2Ghz. 2gb. DDR2 RAM 250gb hard drive D-Link 510N 802.11g Wireless Card with Prism
Router	Linksys WRT-310N	Supports 802.11ngb 10/100/1000 MB. Supports WEP, WPA, WPA2
High Speed Internet	Cogeco	10mbps Download Speed 3 IP Addresses

Software

All of the software required for WPA decryption is included on the Backtrack 4 Linux Live CD. Amongst these applications include Aircrack. Aircrack is a set of tools used in WEP/WPA decryption. Also, the madwifi drivers are required to enable 'injection mode' on the wireless card being used.

Software Title	Description	Availability
Backtrack 3 Live CD	Includes the Operating system and all of the tools necessary for WPA decryption.	http://www.remote-exploit.org/backtrack.html
Aircrack	The main software suite for WEP/WPA decryption.	http://www.aircrack-ng.org/
MadWifi Drivers	Wireless card drivers to enable injection mode.	http://madwifi.org/

Installation & Configuration

The installation and configuration for a WPA decrypting environment is very simple. The Backtrack 3 Live CD contains all of the drivers and tools necessary for WEP and WPA decryption. The Backtrack operating

system can be obtained from http://www.remote-exploit.org/backtrack.html for free. Simply downloading and burning the CD is all that is necessary. Then just place the CD into your CD-Drive and reboot. Within moments a fully configured environment will be loaded for us to use in this endeavor.

Vulnerabilities & Exploits

Although WPA has its vulnerabilities that may be exploited to gain unlawful access to a network, WPA may still be considered to be a highly secure encryption scheme at times. By choosing the right WPA encryption method or an abnormally difficult none-dictionary based password, a user may make decryption of their WPA network unfeasible.

There are 2 modes for WPA encryption, RADIUS or PSK. PSK is hackable, whereas RADIUS is not. PSK (Pre-Shared Key) uses a user defined password to initialize the TKIP (Temporal Key Integrity Protocol). Since the password is user-defined, it creates WPA's largest vulnerability. The TKIP is unfeasible to decrypt since it is created on a per-packet bases. However, upon initialization of TKIP, like during authentication, the user defined authentication password can be obtained.

The Handshake – The WPA handshake was designed to occur over insecure channels and in plaintext. Thus the password is not actually sent across. The password is actually encoded and sent as a primary master key. The only requirement to gain access into a WPA encrypted network is to capture a full authentication handshake between a client and the access point. If a client is already connected, then we can force an authentication handshake through the use of a de-authentication attack (demonstrated later).

Once a full authentication handshake has been captured, we can begin to use a brute force approach to obtain the WPA passphrase.

Test Cases

In order to test the reliability of a WPA encrypted network, we will attempt to gain access to a WPA network that we have set up by using various techniques to obtain the passphrase. Since we will be using our own router, we will configure the router for WPA encryption, and use various passphrases to test the strength of the network encryption based upon the complexity of the chosen passphrase. We will then have a single client connect to the router, and use a separate computer to initiate a de-authentication attack, capture the handshake, and attempt to decrypt the passphrase.

The different passphrases we will select are: small in length (8 characters), large in length (20+ characters), and both dictionary and none-dictionary key results. The table below summarizes the different passphrases that have been selected for the WPA encryption. The table also lists the expected difficulty that will be encountered when attempting to decipher the password through various means. It is assumed that the WPA passphrase is changed between each test case trial.

Passphrase	Type	Expected Difficulty
alphabet	Dictionary Term	Easy
SUPERCALIFRAGILISTICEXPIALIDOCIOUS	Dictionary Term	Easy-Medium
abcdefghijklmnopqrstuvwxyz	Random Letters	Hard
Fdlk8932fdssfjq9ruq234sjflkafd20394asldkfj	Random numbers and letters	Unfeasible

Execution

Overview

The objective is to capture the full WPA handshake authentication, and then use aircrack-ng to crack the Pre-Shared key. Aircrack-ng will attempt to obtain the key using a brute force approach with a qualifying dictionary.

The basic steps that will be undertaken are:

1. Start the wireless interface in monitor mode and on the specific channel as the Access Point. (The Access Point names and channel can be obtained using the 'iwconfig' command)
2. Start airodump-ng on the Access Point channel with filtering enabled to collect the full authentication handshake.
3. Use aireplay-ng to de-authenticate the wireless client. (Optional)
4. Run aircrack-ng to decipher the Pre-Shared key using the authentication handshake.

Trial

Enable Monitor Mode

The first step is to enable Monitor Mode on the wireless interface. Monitor Mode allows us to use the wireless network card to capture and inject packets as required. It is only supported on specific wireless cards, and only if those wireless cards are using a supported driver. In our case, we are using the MadWifi drivers in combination with a D-Link 510n network card.

To enable Monitor Mode, simply stop the wireless interface, and start it with Monitor Mode enabled. The commands to accomplish this are:

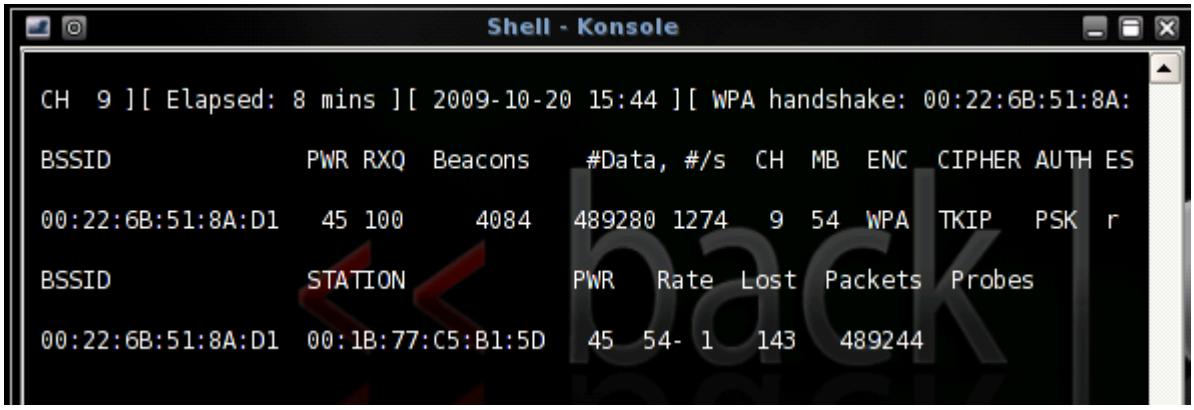
```
airmon-ng stop ath0
airmon-ng start wifi0 9
```

Collect Authentication Handshake

This step is intended to capture the 4-way authentication handshake for the target Access Point we are interested in. This is accomplished using the application airodump-ng.

```
airodump-ng -c 9 --bssid 00:22:6B:51:8A:D1 -w psk ath0
```

Parameters:
'-c 9' is the channel number of the access point
'--bssid 00:22:6B:51:8A:D1' is the MAC address of the access point we are targeting.
'-w psk' is the file name prefix for the file which will contain the IVs.
'ath0' is our wireless interface card



airodump-ng results: Executing the above command returns the following data

Please note the top left information: "WPA handshake: 00:22:6B:51:8A:" This signifies that a handshake authentication has successfully been captured.

De-authenticate the Wireless Client

This step is only necessary to speed up the process of capturing a handshake. To capture a handshake a wireless client must authenticate itself with the access point. This can happen naturally over time, or we can attempt to force the wireless client to reauthenticate with the access point. To force the client to reauthenticate with the access point, we simply send a message the wireless client saying that it is no longer associated with the access point. The wireless client will then attempt to re-connect. The re-authentication is what generates the 4-way handshake that provides us with the information required to break the WPA Pre Shared key.

The output of airodump-ng lists the associated clients with the access point. Simply record the MAC address of a client and input it into aireplay-ng like below:

```
aireplay-ng -0 1 -a 00:22:6B:51:8A:D1 -c 00:1B:77:C5:B1:5D ath0
```

Parameters:
'-0' is the signal to de-authenticate
'1' is the number of de-authenticates to send
'-a 00:22:6B:51:8A:D1' is the MAC address of the access point
'-c 00:1B:77:C5:B1:5D' is the MAC address of the client
'ath0' is the wireless interface

Decipher the Pre-Shared Key

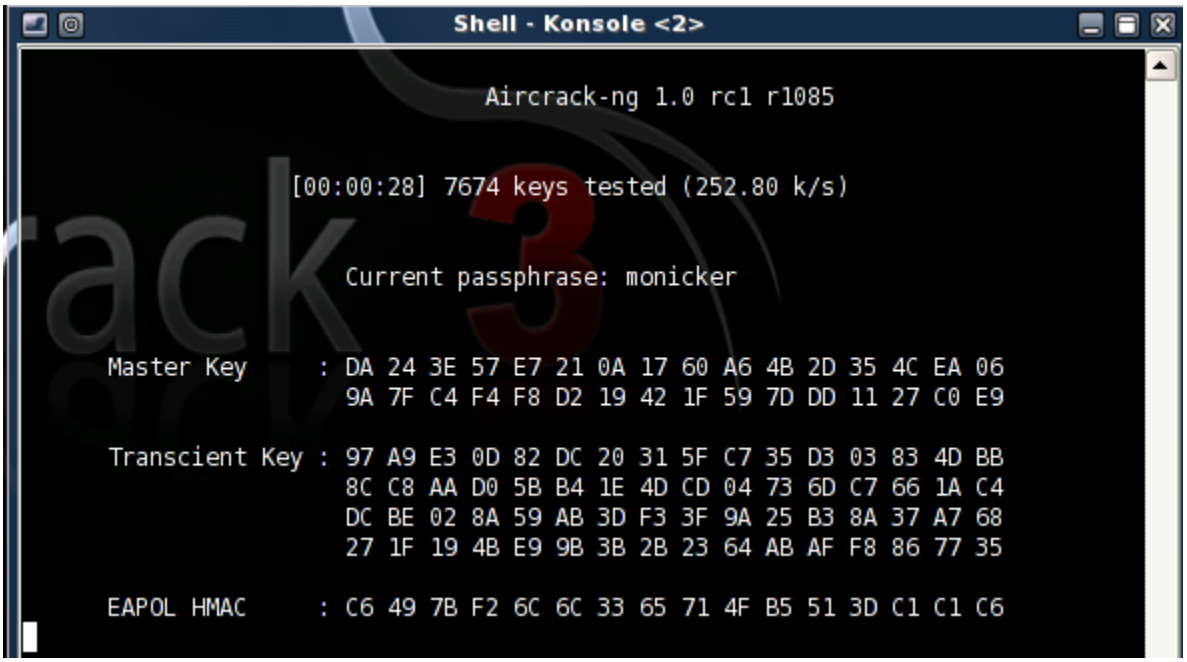
The intent of this step is to actually decrypt the WPA Pre-Shared Key. To accomplish this, a full handshake collection is required (through airodump-ng), as well as a dictionary of words as input. Aircrack-ng takes each word and tests to see if this is in fact the Pre-Shared key. The dictionary used in our test cases were obtained from http://www.openwall.com/wordlists/.

```
aircrack-ng -w dictionary.lst -b 00:22:6B:51:8A:D1 psk*.cap
```

Parameters

'-w dictionary.lst' is the location of the dictionary file

'-b 00:22:6B:51:8A:D1' is the access points MAC address.



Aircrack-ng using brute force dictionary words to obtain the Pre Shared Key.

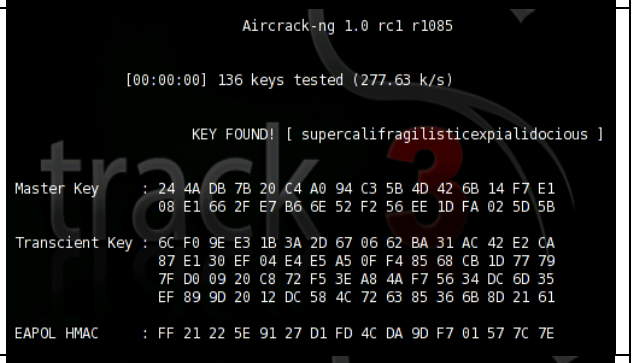
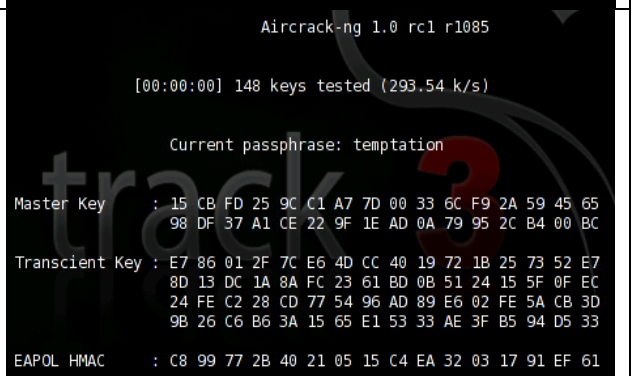
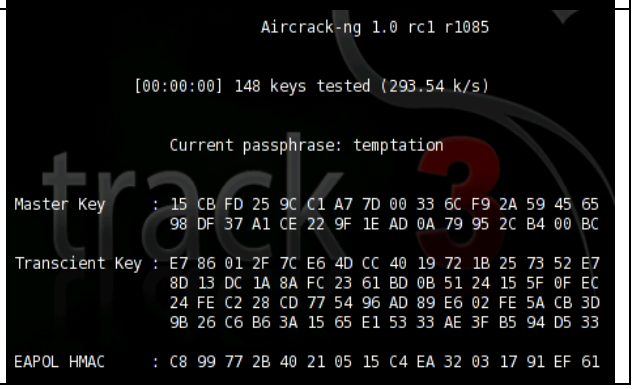
Once a successful key is found, Aircrack-ng will stop checking other dictionary words and report the correct Pre-Shared key.

Results

The four test cases were executed with the new WPA password each time. The primary information collected during each trial was whether the Pre-Shared key was successfully obtained, and the time taken before successfully finding the Pre-Shared key.

The results were as expected. The passphrases that were standard dictionary terms were found promptly. However, the passphrases that could not be found in the dictionary were unattainable. This leads us to believe the total amount of time to brute force the correct password would be unfeasible. The information gathered is displayed in the following table:

Passphrase	Total Time	Results	Output

alphabet	0 seconds	Successful	
SUPERCALIFRAGI-LISTICEXPIALI-DOCIOUS	0 seconds	Successful	
abcdefghijklmnopqrstuvwxyz	-	Unsuccessful – Password Unattainable	
Fdlk8932fdssfjq9ruq234sjflkafd20394asldkfj	-	Unsuccessful – Password Unattainable	

E-Mail Encryption (Protecting Our Privacy)

Overview

In the sections above titled WPA Decryption and Packet Sniffing, we demonstrated the simplicity involved in gaining unauthorized access to a network and then how to sniff traffic along that network to capture sensitive information including e-mail messages and passwords. This section outlines a technique that can be used to secure e-mail messages from anyone that may be invading your privacy. This method is through the use of cryptography. In our test environment, we will run a trial to demonstrate how one can encrypt and decrypt e-mail messages while using the algorithm of PGP and the application GP. The sender will obtain its recipients public key, encrypt its e-mail message, and then send it off to the client. The recipient will receive the encrypted e-mail, and then use its private key to decipher the ciphered text back into plaintext. Through the use of cryptography, one can protect the contents of its e-mail from packet sniffing as well as copies that may be stored on additional servers.

Test Environment

Hardware

To effectively test the e-mail encryption/decryption technique of PGP, we will require 2 computers. One computer is to act as a sender, and the other computer to act as a recipient. Furthermore, we will need an active internet connection. The exact hardware specifics are listed below.

Product Type	Model	Specifications
Laptop Computer	Toshiba Portege R500	Microsoft Windows XP SP3 Intel Centrino Duo 1.2Ghz. 1gb. DDR2 RAM, 160gb hard drive Intel 3945ABG Pro/Wireless 802.11g
Desktop Computer	Custom Build	Microsoft Windows XP Intel Core 2 Duo 2.2Ghz. 2gb. DDR2 RAM, 250gb hard drive D-Link 510N 802.11g Wireless Card
High Speed Internet	Cogeco	10mbps Download Speed 3 IP Addresses

Software

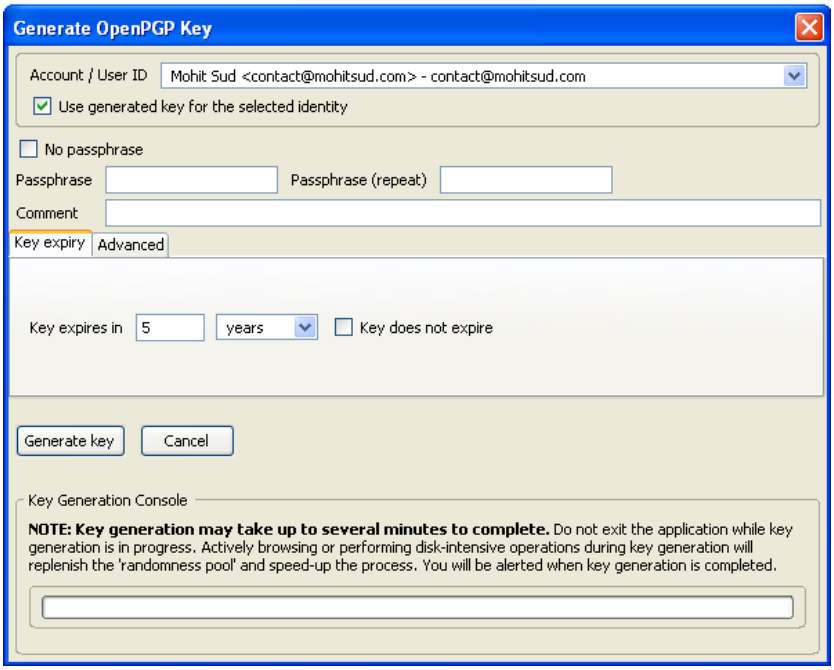
To install and configure both computers to successfully encrypt and decrypt e-mails, the following software is required. Additionally, 2 e-mail accounts are required to test both encryption as a sender, and decryption as a recipient.

Software Title	Description	Availability

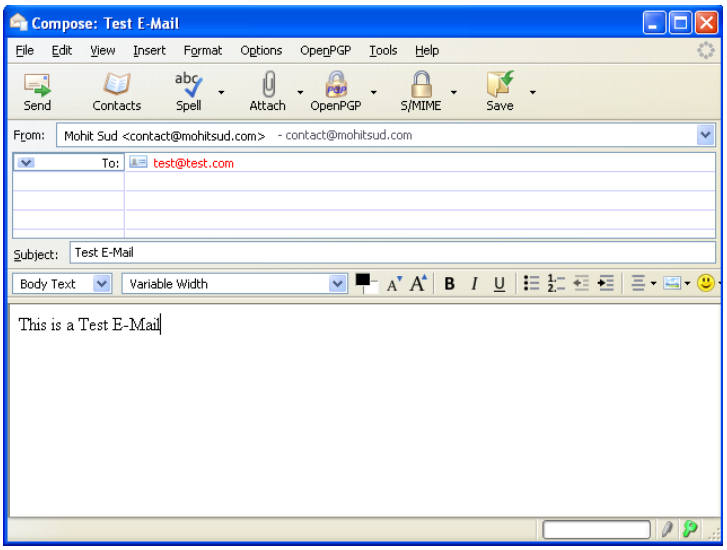
Microsoft Windows XP	Operating system	
Mozilla Thunderbird	Mail client	http://www.mozilla.com
Enigmail Mail Extension	Mail client add on	http://enigmail.mozdev.org
GNUPG	PGP application	http://gnupg.org/
2 E-mail accounts	Two accounts to act as sender and recipient.	Using http://godaddy.com email accounts.

Installation & Configuration

5. Install Mozilla Thunderbird on a computer running the Windows XP Operating System.
6. Configure Mozilla Thunderbird with the information for our e-mail account.
7. Install the Enigmail extension into Mozilla Thunderbird.
8. Install GNUPG and configure Enigmail to find the GNUPG installation files.
9. Use Enigmail to generate your public and private keys. This is accomplished by entering a passphrase, and selecting the 'Generate Key' button.



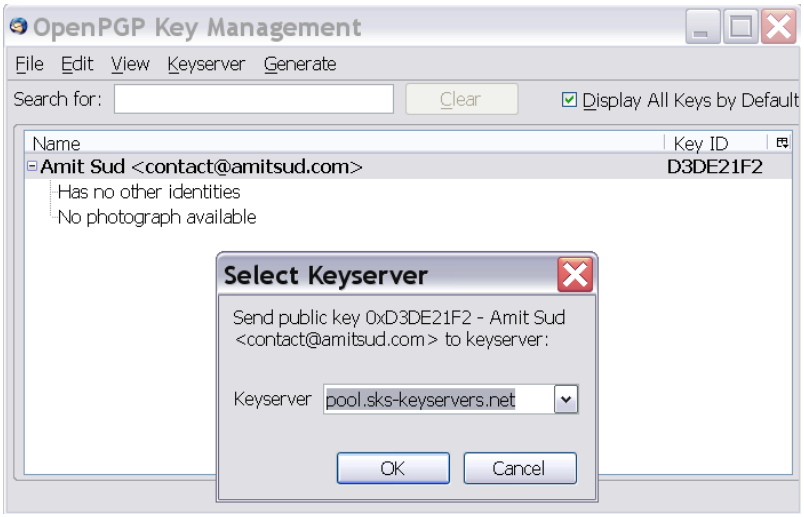
10. Once the pair of keys has been generated, we are ready to send out encrypted e-mails. Compose a new mail as normal, and to encrypt it, simply select toggle the 'key' icon located on the bottom right to the 'on' position.



Trial

Using two separate computer systems, both configured with Mozilla Thunderbird and PGP, I will send an encrypted e-mail message from one computer to the other. The sender will use the recipients public key to encrypt the message. The recipient will then decrypt the e-mail message with their private key.

In order for the sender to encrypt the e-mail message, the sender must first obtain the recipients public key. The sender may acquire the recipient's public key through virtually any communication means available. A few examples include: the recipient may email the key, publish it online, tell the recipient the key, etc. For this example, the recipient will publish their public key online to a Keyserver, and the sender will download the public key from the Keyserver. To publish the key online, Enigmail makes this task convenient. Simply select the 'publish key' feature and select one of the ready-available to store the key to.



The sender may then search the Keyserver for the recipients public key and store that key locally so that for any e-mail sent to the intended recipient, the correct public key will be used automatically.

Once the sender has obtained the recipients public key, the sender may then proceed to sending an encrypted e-mail. The sender composes an e-mail as they would normally, and to turn on the e-mail encryption feature, they would simply select the 'encryption key' icon located on the bottom right of the compose e-mail window.

Results

The sender was successful at obtaining the recipients public key and encrypting the e-mail message. Through this accomplishment, any copies of the e-mail that may be stored on a network server, or any packets gathered by an eavesdropping sniffing on your network will also be encrypted. Thus, those unauthorized individuals attempting to read your e-mail message will only be able to view a jumble of random numbers and characters. The e-mail message is fully secure and still remains private.

The recipient was successful at receiving the encrypted e-mail and decrypting it using their private key. Overall, the PGP trial was successful at keeping the contents of an e-mail message private between the sender and receiver.

Observations & Conclusion

The experiment where we attempted to gain access to a WPA encrypted network through deciphering its passphrase went successfully. We were successfully able to gain access to a network we would otherwise be restricted from by uncovering its confidential passphrase and using that passphrase to authenticate our client with the wireless access point.

The trial where we sniffed the local area network traffic, gathered packets, and analyzed those packets to read the underlying content inside also went successfully. The mission of ease dropping and invading another individuals privacy by reading there e-mail message from a network computer was successfully accomplished..

If we combine the result of both experiments, we can successfully conclude that we are able to gain access to an unauthorized network and sniff through that networks traffic to obtain confidential user information. This leads us to believe that the contents of our e-mail are insecure and exposed to anyone willing to put forth the effort to eavesdrop and read it.

Lastly, the example where e-mail messages were encrypted at the sender and decrypted at the receiver went successful. The information through this means that was exposed on the network was encrypted, and could not be interpreted by anyone except for its intended recipient. This approach proved to be suitable at securing our e-mail messages privacy from eavesdropping packet sniffers, as well as from those wanting to read it when it is copied on a remote server. Although those individuals may be able to see the content of our message, since our message is encrypted, its content cannot be understood unless the private key of the recipient is obtained.

Through these trails and examples it leads us to believe that e-mail encryption is a great approach to undertake if a user desires to have its email messages remain completely secure and private. E-mail

encryption is a simple and easy to perform procedure and works effectively against packet sniffers and those reading potential copies of our e-mails on remote mail servers.

References

The GNU Privacy Guard - GnuPG.org. Web. <http://www.gnupg.org/>.

"Enigmail: Download Enigmail." *Enigmail: A simple interface for OpenPGP email security*. Web. <http://enigmail.mozdev.org/download/index.php>.

"How to encrypt your email - Downloads - Lifehacker." *Lifehacker, tips and downloads for getting things done*. Web. <http://lifehacker.com/180878/how-to-encrypt-your-email>.

"Overview of PGP." *The International PGP Home Page*. Web. <http://www.pgpi.org/doc/overview/>.

"The comp.security.pgp FAQ." *Top Level page for www.pgp.net at cam.ac.uk.pgp.net [08040909]*. Web. <http://www.pgp.net/pgpnet/pgp-faq/>.

"Pretty Good Privacy." *WWW.GAMERS.ORG*. Web. <http://www.gamers.org/~tony/pgp.html>.

"How PGP works." *The International PGP Home Page*. Web. <http://www.pgpi.org/doc/pgpintro/#p1>.

"What is WPA security?" *Belkin : WPA*. Web. <http://en-us-support.belkin.com/app/answers/detail/a_id/34>.

"WPA Wireless Security for Home Networks." *Microsoft Corporation*. Web. <http://www.microsoft.com/windowsxp/using/networking/expert/bowman_03july28.mspx>.

"Cracking_wpa." *Aircrack-ng*. Web. <http://aircrack-ng.org/doku.php?id=cracking_wpa>.

"Openwall wordlists collection for password recovery, password cracking, and password strength checking." *Openwall Project - Information Security software for open environments*. Web. <http://www.openwall.com/wordlists/>.

"Packet Sniffing - Part 1 (wiretaps, protocol decoding and surveillance)." *SuraSoft - Keeping your computer safe! AntiSpyware & Security Information*. Web. <http://www.surasoft.com/articles/packetsniffing.php>.

FrontPage - The Wireshark Wiki. Web. <http://wiki.wireshark.org>.