Wireless Security in Academia

Patrick Trinkle, UMBC
Baltimore, MD 21250
tri1@umbc.edu

December 2006

Keywords:
Wireless Security, Wireless Privacy, 802.11

Abstract

Many universities in the world are providing wireless network access to their students and faculty. These networks, if not properly set up can have a direct impact on the privacy of the students and faculty using these networks. The cost of using an unencrypted wireless network is very high. There are solutions for implementing an enterprise encrypted wireless network, which can handle thousands of users. To fight theft-of-service many schools are implementing an authenticating server system only allowing authorized users onto the network, but this does not solve privacy concerns. This paper is an evaluation of a site survey conducted against the wireless networks of the University of Maryland, Baltimore County, as well as an examination of policy enforcement of students attaching wireless access points to the UMBC residential network.

1. Introduction

Students and faculty at universities have a need for mobile communications, to include wireless networking. A university is the ultimate enterprise wireless network, which must handle thousands of users, who often arbitrarily change devices. With this environment, implementing a wireless network leaves two basic options: a network, which is non-secure, but easy; or a network, which is very secure but difficult to use. The University of Maryland, Baltimore County currently implements a wireless network with an authenticating backbone to control unauthorized use of network resources. This approach handles authentication of users, but neither the integrity of the data, nor the privacy of the data are considered.

Most wireless networks follow the IEEE standards of 802.11, which are broken into 802.11a, 802.11b, and 802.11g. The primary difference between 802.11a and 802.11b and 802.11g is that 802.11a runs at 5 GHz frequency, whereas both 802.11b and 802.11g run at 2.4 GHz [7]. The 802.11b standard has a data rate of 11 Mbps, whereas 802.11g has a higher rate of 54 Mbps. UMBC implements both 802.11b and 802.11g systems. Devices supporting 802.11g also support the 802.11b standard. These wireless devices are using radio waves to transmit their identity and their information. As is true with radio waves, anyone with the proper equipment can receive the transmissions.

Traditionally university networks have been relatively non-secure for policy reasons and the constantly changing needs of students and faculty. As such, university networks make excellent targets in many various cyber threats, to include wireless attacks. The wireless attacks targeting universities and other enterprise environments include attacks seeking to exploit personal non-public information (NPI). NPI can include credit card numbers, email addresses, screen names, usernames and passwords, and web surfing trends. Another wireless attack targeting enterprise systems is an attempt to gain surreptitious access to an internal network from a wireless access point. Once access is gained into a network, the entire network is vulnerable, to include file servers, email servers, and any confidential data stored on the network. The range of

802.11b and 802.11g radio frequency transmissions is quite large, therefore wireless attacks are feasible from great distances. Great distances are not necessary for an inconspicuous attack, for either capturing of data or gaining unauthorized access. An attacker could sit in his or her car in a parking lot next door, or sit in a coffee shop across the street. Without a security mechanism in place to encrypt the data frames transmitted, an attacker with a COTS wireless network card can read the raw packets as the computers communicate wirelessly. Only with a wireless card that is in rfmon (radio frequency monitor) mode can this be done. A wireless card in rfmon mode is in promiscuous mode. In promiscuous mode, the wireless network card listens for frames and captures them passively. This mode allows a network card to capture all data in range on a specific channel. Packets captured via a wireless network card and software are stored in a pcap file. The pcap file can be loaded into a packet analyzer, which allows an attacker to filter the packets easing the search for information. Filters can pull Hypertext Transfer Protocol (HTTP) or HTTPS traffic, File Transfer Protocol (FTP) traffic, telnet traffic, and Simple Mail Transfer Protocol (SMTP) traffic from the very large pcap file.

To demonstrate the feasibility of this attack, I will use an open source tool for both packet capturing and packet analysis. In addition, all hardware used will be easily obtainable by the public. The packet-capturing tool is Airodump 2.3 in the Aircrack 2.41 suite by [1]. Packet analysis is performed using Ethereal 0.99.0 [2]. The wireless network card is a Cisco Aironet 350 with a 5-dBi range extending blade antenna.

UMBC also has a policy allowing students to attach wireless access points to the residential network as long as security measures are in place. This policy is addressed in the following aspects: is this a good policy; how to enforce this policy; would it be easier to provide another option for students; is there an effective way to educate students in securing a wireless network.

Common wireless access point security features include, a crypto string used for encrypting the packets, not broadcasting the access point's SSID, and a MAC address access control list (ACL). There are many implementations of the crypto string, also called a Pre-shared Key, such as Wired-Equivalent Privacy (WEP), Wi-fi Protected Access Pre-Shared Key (WPA-PSK), and WPA2. The weaknesses of WEP are easily exploitable as outlined in [3]. The most common exploit to WEP is the weak IV attack, whereby with enough collected weak IVs the key is guessable. With enough captured packets, WPA-PSK is vulnerable to a similar attack as guessing the key is relatively trivial. In enterprises, the idea of a key every device must know is burdensome, especially if the key rotates regularly. Not only the system administrators, but also the end-users feel this burden. Often burdensome security is neglected.

A War-Driver is someone who locates and publishes the locations of wireless hot spots (or access points) onto the internet or with a physical marking, such as chalk (war chalking) [7]. Netstumbler 0.4.0 [6] is used to locate wireless hot spots in a physical location. Attached to a GPS device, this software will automate the mapping of hot spots onto a map. War-Drivers can use this generated map to publish more easily their findings onto the internet.

2. Background

UMBC's wireless network must handle only a few hundred users at any time, but must support the potential for thousands rotating on and off the network itself. Authenticating this workload of users is no small task, but simple authentication servers without many problems conduct the handling of this process. UMBC uses a Vernier product to implement this. The problem inherent to UMBC is the privacy of students using the wireless network. These authorized users can have their data stolen out of midair. UMBC wireless access points support communication on four channels: 1, 3, 6, and 11 at both 802.11b and 802.11g standards and data-rates.

The problem of privacy in wireless security has always been present, because there is no real physical barrier. Data integrity is also a real concern with data transmitted over the air, but this is not addressed in full by this paper (airpwn?). Client authentication is also a problem because of attacks like replay, and man-in-the-middle. There exist a few current solutions which provide a certain level of authentication, primarily 802.1x. Data privacy or confidentiality, which is the third facet of data security, is a primary concern in this paper.

A student or attacker could easily inconspicuously capture traffic from the wireless networks and build a model of user surfing patterns; build a database of usernames and passwords; build a database of non-public information to include anything that might go in the clear—such as social security numbers, credit card numbers, home addresses, et cetera. The motives of this attacker might be malicious in so much as to use the data against the students directly, or the attacker could be attempting to turn a profit by selling this information to spammers or other individuals seeking such information, possibly identity thieves.

Another problem facing UMBC is the enforcement of the policy regarding the attachment of secured wireless access points to its residential network. The policy states that some method of encryption must be in place, such as WEP. This encryption and authentication must be in place to actively prevent theft of service and privacy of data. If this is a solution then another problem asserts itself—how to enforce such a policy? There is no current method for polling a network for wireless routers and determining which are without encryption. Therefore, students and faculty users are trusted to follow the letter of the law. This trust cannot exist in a secure networking environment.

If an attacker gains access into the residential network through a student's open wireless access point, then the student has opened the entire residential network to the attacker. This attacker now has a jump point throughout the network, to include potential exploitation of other computers on the network.

3. Methods

Windows requires WinPcap to handle link-layer network access [4]. This software is required for Ethereal to operate properly. Airodump also had some driver requirements for proper monitoring and capturing of network traffic. Airodump can run in two modes, one targeting a single channel, and the other is channel-hopping mode. The channel hopping mode jumps between all channels, 0 through 14. To monitor all the UMBC channels, the software runs in the channel-hopping mode. Since UMBC does not implement encryption of data, the packet capture or pcap file is in plaintext and therefore no decryption is required to read the packet contents with Ethereal. If UMBC was implementing WEP, there is another tool in the Aircrack suite that can accurately guess the key after enough packets are captured, thus allowing the decryption of the captured packets. There are also other open source tools, which implement the cracking of WEP. The operating system of the capturing machine is Microsoft Windows XP. There exist other software solutions for attacking or passively capturing wireless packets for Linux [5].

The initial attack against UMBC's wireless network was a passive packet capturing attack. Airodump was set to channel hop. The collecting sessions conducted were for approximately an hour in length over the period of 5 days, a Monday through that Friday. The collections were conducted between 11:30 am and 12:30 pm. The laptop was kept in plain sight in the Commons building on the first floor. This behavior of passively capturing wireless packets is undetectable and rather inconspicuous. To the casual observer the passive collection appears as merely surfing the internet or using the laptop for educational purposes. Passive packet collection leads to neither degradation of network services, nor does this attack provide any trail—therefore it is impossible to detect via the network.

The staging of the second passive capturing attack against UMBC was very similar. The only difference was the selection of a set channel. This provides for a considerably more consistent data stream. In other terms, it is a lot like trying to listen to a song on the radio while changing the station 10 times a second. Of the channels UMBC uses with its wireless network, I chose channel 11. The choice of channel 11 was not arbitrary. It appeared as though more access points and laptops were using channel 11. As previously, the laptop collected for approximately an hour for 5 days in a row, also during lunchtime.

The third part of this site survey was an assessment of the adherence to the UMBC policy concerning wireless access points attached to the residential network, resnet. The tool used to conduct this survey was the network reconnaissance tool Netstumbler 0.4.0 [6]. When attached to a GPS device, this tool can literally aid someone in mapping wireless access points. I did not use a GPS, just a map of the UMBC campus. From this map I selected various points to stage the network discovery attack. The choice of the locations was based on their proximity to dormitories and other residential facilities on campus. I either parked at such location, or found a bench and sat inconspicuously. At each location, a fresh discovery file was used with Netstumbler to keep the data accurate and separate between locations. Eight locations gave a relatively complete view of the residential areas. It was not necessary to cover every possible location, to find every wireless access point. The purpose was only to find whether students were all adhering to the wireless policy. Applying Filters to the scanned view of the area, allowed discrimination of information, such as displaying access points using encryption and those not. During these collections, any network whose SSID was tsunami was ignored as part of UMBC's attempt to bring wireless access to the residential areas. It was also noted which encryption the access points utilized.

Interpretation of the pcap files from the wireless capturing was done via Ethereal. Once the file was loaded three things were recorded: the duration; the number of packets; the total number of bytes. Then filters were applied systematically to each file and the results recorded. Filtering of FTP traffic is desirable because passwords are sent as plaintext. This filter displayed only packets involved in ports 20, and 21. Telnet also suffers from limited security, in that its passwords are also sent over in the clear, therefore filtering against port 23 was useful. Another interesting protocol is SMTP, over port 161. This can contain email addresses, email contents—which may be private, et cetera. Another interesting protocol to examine is SNMP. Retrieving community strings in Simple Network Management Protocol can grant certain levels of access to the network infrastructure. The most interesting protocol observed was HTTP and HTTPS, which is HTTP with Secure Socket Layer (SSL). Non-secure HTTP communications use port 80, while 443 is for HTTPS [8]. Once a filter for web traffic has been utilized, data posted to websites can be read. Also, searches can be more quickly and fruitfully completed for terms like "password=," "passwd=," and "pass=." Those query strings were used to parse the web traffic for passwords stored in Cookies as well as in POST and GET requests and in-line hypertext markup language (HTML). Websites often use Cookies to store personal information, to include the username and a hash of the user's password. Sometimes, a website will store the password in plaintext.

4. Results

For all analysis of usernames and passwords, those sought out were strictly plaintext. This disregards passwords potentially stored in cookies encoded with base64. Narrowing the search down to plaintext passwords allowed for faster searches into the massive data set, and as such more immediate results. If no plaintext passwords were found; a second sweep of the data set would have sought out base64-encoded strings; which could reveal usernames and passwords. The pcap files were loaded into Ethereal, and filters applied as described in the Methods section.

The total number of packets captured was 7,897,969. 2,257,276,594 bytes over 9:59:39 were captured from both passive listening attacks. With this much data it proved difficult to perform a deep analysis on the traffic in such a short time frame. Although with only shallow analysis, defined as seeking out the obvious, many pieces of information were captured. Had the attacks run over a longer time, substantially more data would have been retrieved, and with it substantially more NPI.

Every day of collections found massive amounts of American Online Instant Messenger (AIM) [9] Traffic. This traffic did not include passwords in plaintext, as AIM encrypts passwords and sends the md5 hash to the server for authentication. The filter applied to the data set was a filter for port 5190. When a user logs into the account, the server sends the client a status message indicating which "buddies" are currently online. When the client requests the "buddy information" for a particular user, the server sends this information to the client. An AOL screen name can be attributed to a particular IP address, which—only while the machine is logged on in real-time—can be attributed to a machine name. Other tools can enumerate more about this machine, and once such process is completed the attacker has linked a person's identity to his or her computer, and to all their "buddies." In addition, the client and server communicate user conversations and such in plaintext. In short order, after examining this filtered traffic, conversations with telephone numbers were found, buddy information including such as well. Tapping someone's telephone is illegal because of the right to privacy, therefore listening to their conversation over the computer should fall under the same reasoning [10]. The data set included hundreds of screen names, and dozens of conversations and "buddy information" replies.

On five days, the data set included FTP traffic. The FTP traffic revealed the IP addresses of a number of FTP servers, which allowed anonymous login. The packets showed the clients connecting to the server, logging in as anonymous and displaying their passwords in plaintext. In this case, the password was unimportant. What is important from these results is evidence that such data can be captured without any sophisticated equipment or particular knowledge of the FTP protocol.

The telnet protocol is a method for console level access to another computer. There was minimal telnet traffic collected.

Only on 1 day was SMTP traffic collected. Collecting SMTP traffic can reveal the contents of emails, as well as provide email addresses. Locating SMTP servers is also useful as they are often exploitable, due to a non-authenticating protocol [11].

Simple Network Management Protocol is used primary to "getting", "setting", and "triggering" certain items in the network infrastructure, to include routers and switches. Capturing this traffic can divulge information on the configuration of devices on the network. This traffic was not found over the wireless network during the attack sessions. It is feasible that all such transactions are handled entirely over the wired network or by physical consoles to a majority of the core devices.

Filters were applied to display only traffic over ports 80 and 443. This restriction allowed for efficient searches, as described in Methods. HTTP traffic can reveal many interesting pieces of information. When a user fills out a form on a website, the data is sent to the website via either a GET or a POST request. These requests are intercepted and easily interpreted. Many sites either do not incorporate SSL for important transactions or do not properly use it. Another aspect of analysis of HTTP traffic falls into the category of user analysis. Attackers can develop student internet surfing patterns once enough data is collected. This can provide insight into targeting UMBC students with specially crafted spam. Aside from the attacking NPI and traffic patterns, certain packets contain information for storage on a Cookie. Cookies are a method for a client computer to maintain state on a website. Other packets contain the contents of the cookie as is it transmitted to the server. Cookies contain user data, which the site requires to identify the user. This identification of a user to a site and a user to a computer can allow for a record of this user's data and behaviors. Often users follow the poor password usage, and by this, they will use the same authentication for multiple websites and services. In both collection phases usernames and passwords were recovered. Twenty-five total usernames and passwords were collected. The three strings defined in the Methods section were used to locate packets with passwords. Since no base64-encoded packets were examined for the possibility of passwords, there exists the potential that many pieces of information were overlooked.

Overall, a considerably negligible amount of damaging data was collected during the short experiment. Placed into a larger context however, the amount of collectable data is a strong privacy concern for UMBC students.

The site survey conducted on campus was comprised of both war driving in a vehicle as well as on foot. During collections, I was never approached by anyone inquiring as to my business, to include campus police or other students.

The sites were chosen based on their proximity to dormitories and on campus apartments. At each site, a substantial number of wireless networks utilized any sort of data encryption or access controls. An average percentage of open wireless networks over all eight sites is 34%. For more detailed information see Table 4-1. This demonstrates that approximately 1/3rd of wireless networks found were open to the outside without any protection. This number only represents the percentage of those found and the survey did not find all networks. Those networks found that employed a security mechanism used WEP.

5. Discussion

Privacy of data in transmission is critical. Authentication of users on the network is also critical. The results the wireless passive collection attack produced were frightening and predictable. There is a clear lack of privacy and a self-evident need. The data collected revealed traffic patterns of students, their usernames and passwords, their conversations, their friends, and their identity. With the heavy burden of knowing many different passwords, users will often use the same password for multiple authenticators. This idea allows for the assumption that although a password for one website was found, some users will also use this same password for other services, such as email and AOL Instant Messenger. There are solutions that handle the data confidentiality and user authentication separately. One such solution is using WPA with a pre-shared key, and the current authentication method. This would however require all users to know the pre-shared key, which should be rotated regularly to maintain effectiveness. This task is obviously a huge burden and rather infeasible in such an enterprise environment. Implementing WEP2, which is not susceptible to the known WEP attack, still brings the burden of key rotation [12]. A viable solution to UMBC's wireless network needs to be secure, quick, easy to administer, and cover the requirements of data encryption and user authentication. Administration has to include adding users quickly, as well as revoking privileges manually as well as on a time based criterion. Public Key Infrastructure (PKI) is an interesting solution, which can vary through implementation details. Essentially PKI falls into the idea of a public and private key used to authenticate a user and encrypt a symmetric key which is then used to encrypt traffic [13]. PKI carries problems in a large infrastructure, such as key establishment, key rotation, and key revocation. The key establishment problem is such that thousands of keys can be pre-generated with potentially many wasted, or many generated on the fly. Software can handle the key generation, but once the keys are generated, the clients need to install the keys to use the services. In many cases, the administrators may need to handle this, because many computer users may not understand how to do it. Key rotation obviously suffers from large problems, unless there is a mechanism built into the authentication system that automatically replaces the old certificates with new ones. When a user is no longer allowed on the network, the user's keys need to be invalidated and not reused. There are many cases where such an action is routine. Such instances as a guest pass on the network, where such pass expires after 24 hours. A user may violate the network usage policy and is required to follow some course of action to regain admittance to the network.

There is an interesting solution to the problems of data confidentiality (encryption) and user authentication. This solution was described and implemented in [14]. To handle enterprise networks, users bring their wireless device to an enrollment station. The user runs a 2 step program, while connected to the enrollment station through a location limited channel. This location limited channel prevents eavesdropping on the enrollment key exchange between the wireless device and the enrollment station. Once the wireless device has the required information, it can continue communicating over the wireless network. The NiAB solution handles key rotation automatically. Key revocation is handled very simply and simple augmentations can be made to allow for expiration of keys. The simplicity of device enrollment eases the burden on the administrators, although, all devices must manually attach to the network. This can be an overwhelming task, unless there is a mechanism in place, whereby users can schedule their enrollment into a timeslot. Enrollment itself takes less than a minute. The enrollment station would need to be manned, as users could be required to show some form of identification. The details are not important, only the notion that it would be feasible to have enrollment times or appointments if necessary. Once certificates are granted, there can be a life that lasts the reasonable timeframe for their usage. At the university, the keys could easily last a semester. The first couple of weeks of the semester, the enrollment station can be manned between a certain set of hours. UMBC already dispenses a CD containing useful software for interacting with UMBC resources. The enrollment software could be added to this CD and/or kept on hand at the enrollment station, as the user only needs the software initially. In the paper the location limited channel implemented is an infrared (IR) port. As these are uncommon, and it is only a detail, this same principal can be implemented with a Universal Serial Bus (USB) cable, or a crossover Ethernet cable. The use of either of these makes it a considerably more practical solution in the university environment as well as eliminating the potential for compromising the security of the IR transmissions. Either this solution or something similar in principle can handle a university's authentication and encryption needs. If another solution is found it also must address usability. Usability is paramount to handling the variety of users and the number of users to attach to the network.

Handling users attaching wireless access points to the UMBC residential network occurs. Attaching wireless access points can lead to serious problems. These problems include, while not limited to, theft of service, tunneling of illicit materials, and non-authenticated access to the UMBC residential network. This non-authenticated access allows a device to enumerate other devices on the network, and potentially exploit them. This access point therefore acts as a backdoor to the network. To avoid these problems, UMBC instilled a policy whereby students were allowed to implement access points as long as a certain measurable security was in place. UMBC's Office of Information Technology (OIT) also released documentation to help set up certain devices. Such a policy is acceptable, but difficult to enforce.

A method for enforcing the wireless access point security policy can entail a physical site survey. A rather accurate and complete site survey would not require much time and can be conducted periodically. Lightly trained OIT staff can handle the physical site survey. Once the surveyor associates to the wireless network, the IP address of the device reveals the identity of the owner. This method is straightforward and effective. Each staffer can be assigned a building or set of buildings, and in short order have a rather accurate map of the wireless access points present—including which are implementing security. Another method of enforcement sets the responsibility onto the Residential Assistants (RA). UMBC Residents can be required to register such devices with OIT through the RA. This method requires trust of the residents and the RAs.

Another aspect of security is the user. Active education to the users of wireless networks about the security risks involved, could promote an environment of educated network security practices. As students enroll in the university they are taken to different events—it would not prove difficult to bring wireless security education in as an event of sorts. Many users attach wireless access points out of convenience without understanding any of the aspects of what consequences and responsibilities fall onto them. Users of the university network are responsible for all traffic over their access to the network. Therefore, users may not be aware that other users can use their access to violate the UMBC internet usage policy. This course of action can temporarily revoke the legitimate users' access to the network. Educated users are more likely to act in an educated manner, and in this situation properly set up their wireless devices.

Regarding the policy itself, it could be more effective by specifying a minimum-security posture. This minimum-security posture can detail the usage of 128-bit WEP encryption, although WEP is ultimately ineffective. This policy also needs to be maintained as the prevalence of newer wireless technologies becomes available, to include WPA2. Modern devices support WPA, and therefore this can be considered a more secure minimum-security posture.

6. Conclusion

This paper addressed two major aspects of wireless security in a university enterprise. Firstly, whether there was a privacy problem with unencrypted wireless traffic, and secondly the usage of wireless access points attached the university's residential network. This paper also considered solutions to both the problem of privacy and policy.

A large security gap was discovered in the usage of an unencrypted wireless network at UMBC. Although the concept of an unencrypted wireless network not being secure is old and addressed, UMBC still uses an unencrypted network. Large enterprises like universities fall into this category, wherein convenience vs. cost plays a part in the necessity of wireless security. After a rather short passive collection attack against the UMBC wireless network, it became clear that students' privacy was nonexistent. Over a longer time period considerably more information can be collected and used against UMBC students, faculty and staff. In addition, with many open- source wireless passive collection tools available, and the notion that passive wireless collection is near undetectable; the threat of a passive collection attack is imminent. Attackers are already using wireless attacks against corporations. An attacker can UMBC can collect gigabytes of user traffic with relatively little effort and completely inconspicuously. This data can then be quickly parsed for NPI. UMBC and other university enterprise networks need to consider the use of an encryption and authentication mechanism. The model encouraged by this paper is a modified NiAB enterprise implementation [14]. Other solutions are acceptable as long as they consider usability, data confidentiality (encryption), and authentication.

The policy regarding wireless access points attached to the network was examined through a physical site survey. This site survey evaluated the effectiveness of the policy as well as the potential capabilities for the enforcement of the policy. It is feasible to enforce the policy although it is encouraged the policy is altered to detail specific security requirements. Enforcement of the policy can take three versions, one that the RAs handle the responsibility, and the other were the OIT staff does a physical survey seeking violators. The third enforcement method lies in actively educating the users of the network of the risks of open wireless. The documentation detailing the usage of security in wireless could be more easily available.

Acknowledgements

This research could not have been successful without the cooperation of UMBC. Also notably Dr. Trinkle for aiding in grammar and style.

References

[1] Aircrack 2.41, 16 October 2006. http://www.aircrack-ng.org

[2] Ethereal 0.99.0, 4 October 2006. http://www.ethereal.com

[3] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In Eight Annual Workshop on Selected Areas in Cryptography, August 2001.

[4] WinPcap 3.1, 4 October 2006. http://www.winpcap.org/

[5] Kismet, 4 October 2006. http://www.kismetwireless.net/

[6] Netstumbler 0.4.0, 4 October 2006. http://www.netstumbler.com/

[7] Wireless LAN Security, 802.11/Wi-Fi Wardriving & Warchalking. 4 October 2006. http://www.wardrive.net/

[8] Port Numbers, October 2006. www.iana.org/assignments/port-numbers

[9] AOL Instant Messenger, 15 October. http://www.aim.com

[10] 2511. Interception and disclosure of wire, oral, or electronic communications prohibited. 4 October 2006. http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002511----000-.html

[11] Simple Mail Transfer Protocol, RFC 2821, 4 October 2006. http://tools.ietf.org/html/rfc2821

[12] W. Arbaugh. An Inductive Chosen Plaintext Attack against WEP/WEP2. May 2001, http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm

[13] What is a PKI?, 4 October 2006. http://www.entrust.com/pki.htm

[14] Dirk Balfanz, Glenn Durfee, Rebecca E. Grinter, Diana K. Smetters, and Paul Stewart, PARC.

Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute. USENIX Annual Security Conference, pages 207–222. USENIX Association, 2004.

Table 4-1

Site	**OPN**	**ENC**	**%OPN**
1	3	8	27.27
2	5	16	23.81
3	4	8	33.33
4	7	11	38.89
5	8	11	42.11
6	2	3	40.00
7	4	6	40.00
8	3	6	33.33