Secure Wireless Gateway[1]

Austin Godber and Partha Dasgupta
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ, U.S.A.
{godber, partha}@asu.edu

ABSTRACT

Wireless LANs (WLAN), using the IEEE 802.11b standard, have been shown to be inherently insecure. Given the widespread use of this type of WLAN for public and corporate access, it is important to have an "idiot proof" method for securing WLAN from hacking, sniffing, and unauthorized access.

In this paper, we present a simple solution using IPSEC that provides an inexpensive, easy to implement, wireless gateway, and an access point that is secure. The client configuration involves no additional software, and the simple steps needed to configure a client are provided using a captive portal. Thus, the gateway is designed to minimize the intrusion to the end user, will only be slightly different from using a standard wireless network, and will require no additional software or hardware.

Categories and Subject Descriptors

C.2.1 [**Communication Networks**]: Network Architecture and Design – *Wireless communication.*

General Terms

Security

Keywords

Wireless LAN, IPSec, 802.11b

1. INTRODUCTION

Wireless LANs (WLAN), using the IEEE 802.11b standard, are being deployed ubiquitously at a remarkable pace. Low cost, ease of operation, platform independence, and product variety make them appealing to everybody. Corporations, as well as individuals are deploying WLANs for a variety of reasons. WLANs are being used as general, easy to deploy extensions of local area networks. They have been used to link remote LANs from miles away, and to provide wireless Internet connectivity to the public in many metropolitan areas. The versatile nature of 802.11b has also made it an ideal target for "hacking."

Regardless of how the WLAN is being used, a standard WLAN implementation will always suffer from problems inherent in the underlying 802.11b standard. 802.11b is a broadcast media, where the interface to the LAN is often a bridge (access point). Using a bridge results in all network traffic on the local network segment being broadcast to anyone who cares to listen. The range of a WLAN is supposed to be limited, but the use of special equipment (high-gain Yagi antennas) easily extends the eavesdropping distance to miles. When a router, instead of a bridge is used to bridge the WLAN to the LAN, only the wireless traffic gets broadcast, but this is just marginally better from a security point of view.

1.1 WLAN Insecurities

Not only can one eavesdrop on the network traffic, but there is also no effective mechanism to authenticate users of the wireless segment. Being a broadcast media, one cannot rely upon any physical security to control access to the network; therefore, a reliable authentication mechanism is essential.

The 802.11b standard tried to address both the eavesdropping and authentication problems by using an encryption protocol called WEP (Wired Equivalent Privacy). WEP is based on the RC4 stream cipher, which uses a static 64 or 128 bit key. A theoretical attack [1] against WEP was published in the summer of 2001 and an implementation [2] and open source exploits, like WEPcrack and Airsnort, quickly followed. The published results indicate that the WEP key can be extracted using a ciphertext-only attack, i.e. just by eavesdropping, regardless of the length of the key.

In addition to WEP, wireless base station manufacturers attempt to provide client authentication by filtering traffic, based on IP addresses or MAC addresses. Both of these addresses are low layer mechanisms that are capable of being spoofed; thus, providing no real security. It is relatively easy to steal MAC addresses or IP addresses by eavesdropping and using them later to gain access to the network.

Furthermore, the majority of the installations of WLANs use default settings, i.e. no WEP key or filtering is enabled. This provides a "backdoor" entry into a network, behind all the

WiSe'02, September 28, 2002, Atlanta, Georgia, USA.

[1] This research in partially supported by DARPA/Spawar, AFOSR and NSF.

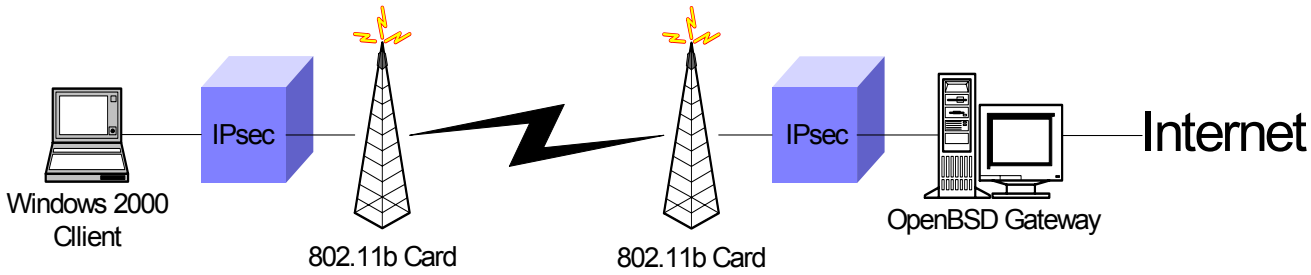

Figure 1 - Wireless IPsec Gateway

firewalls, leaving a gaping hole in all the security provisions that are deployed elsewhere in the network.

1.2 Securing the WLAN

Despite its shortcomings, 802.11b is still an innovative and compelling technology. One of the most interesting applications of 802.11b is that it can be used to provide a public wireless gateway to the Internet. Networks, like this, are appearing in many major metropolitan areas around the world and there are many projects and groups dedicated to expanding these networks. Commercial providers use 802.11b to serve customers at cafés, airports, hotels, and shopping centers.

The majority of the commercial providers provide security and authentication, using the "captive portal." As soon as a new user connects to the WLAN, a router behind the access point, using MAC filtering captures all packets from the users machine and sends them to a particular server; regardless of the destination. The user cannot have access to the Internet, until he or she opens the web browser.

The first http request from the browser reaches the designated server and the user sees a web page prompting him/her for a username and password or for a credit card number. After the user provides the required authentication information, the MAC filter is disabled and all the traffic from the users' computer is allowed to flow normally. This mechanism has some serious deficiencies. First, if the captive portal does not use SSL to get the authentication information, passwords or credit card numbers can be stolen by an eavesdropper. Second, after the authentication is completed the MAC address of the authenticated user can be stolen and be spoofed to gain unauthorized entry (MAC addresses on most cards are software updateable). Third, the communication can be overheard by a third party. Finally, the above attacks, denial of service and hacking can be easily launched.

2. MOTIVATION

In this paper, we describe a method and its implementation that mitigates much of the security problems of 802.11b based WLANs. Our major design goals are ease of use and needing no special software to be installed on the client machine. The approach will allow any person to access the WLAN, and proceed to use it, with minimal hassle. Both registered users and "free" access are supported, and we can provide varying levels of quality of service; depending on the user profile. Thus, the target of this method is the commercial WLAN service provider; although it is just as useful in other applications, such as office, campus or home networks.

A public wireless gateway, like this, faces a unique set of challenges. Providing public Internet access to unknown individuals exposes the provider with two major risks. The first risk is that since the provider has only a finite amount of bandwidth; there is the possibility that the public users may completely consume that bandwidth, leaving the other users without reasonable service (denial of service). Clearly, this situation needs to be avoided. The provider is generously providing a free service, but it is unlikely that he would want to do so at the expense of not being able to serve all users fairly. The second risk is the possibility that an anonymous users may be taking part in activities, which could result in the provider's ISP shutting down service, for example sending spam email, attacking other computers, and participating in DDoS attacks. Third, a malicious user could capture sensitive information, such as passwords, files, and other data by eavesdropping on other users' communications, using the same WLAN.

The security of corporate LANs and privacy of home networks depend on the administrator's ability to reliably prevent unauthorized access and to avoid the monitoring of traffic. Security through obscurity does not provide adequate protection; even from casual network abuse. Just like the 80's, when modem dialups were attacked using "Wardialing", and the 90's had port scanning of new Internet hosts, the early 21st century has its own hot new communication technology: 802.11b. With new technology comes a new form of scanning and 802.11b scanning is known as "Wardriving." It has become just as effective as its predecessors at sniffing out vulnerable networks. Anyone equipped with a Laptop, a wireless card, and the right software can drive around town, looking for vulnerable wireless networks.

Wardriving has become so sophisticated that users with GPS receivers have generated maps of WLANs and share these maps on web sites devoted to sharing WLAN information. Since so much information about a WLAN can be gathered by an individual and an individual can share the collected information so easily, a WLAN will not go unnoticed for long. Therefore, reliable security is necessary.

A mechanism to securely authenticate users would form an important first step in addressing these unique problems. In addition to client authentication, a public wireless gateway should provide bandwidth shaping, a mechanism for securely controlling client sessions, and all but the initial authentication should be done transparently from the client's point of view.

To summarize, the features we provided in the approach are described below:

1. A method of authenticating registered as well as non-registered (casual) users, without divulging sensitive information.
2. Providing different levels of service for authentic users (paying customers) and casual users.
3. Providing secure (eavesdropping-free) channels for all users.
4. The use of simple, off-the-shelf components (hardware and software) to enable the above.

3. APPROACH

A successful implementation of a wireless gateway would not require the client to install any additional software, given that the client is using a standard, off-the-shelf, business operating system. The mechanisms employed to interact with the gateway should require nothing more than a few simple steps for authentication. This method should only be slightly harder to accomplish, than using 802.11b under normal circumstances.

Ideally, a wireless gateway would have the following components: an encryption and authentication module, a web and DHCPD module, and a routing and bandwidth-shaping module. These modules could coexist in a single gateway machine or, if done carefully, they could be separated into different machines.

The encryption and authentication module would be responsible for authenticating each client, using some cryptographically secure authentication method. It would also be responsible for establishing an encrypted tunnel between the client and the gateway machine.

The web and DHCPD module would handle network services, like web and DHCP. The main goal of the web server would be to facilitate the initial client authentication using the *captive portal* mechanism, described above.

The basic protocol for an initial client authentication would be as follows: a user would start his machine and it would make a DHCP request. The gateway would give the client a lease. At this point, since the client has not yet been authenticated, the gateway only permits the client to communicate to the gateway itself, all other traffic is denied. The user of the client machine will try to establish an Internet connection, and at some point it will attempt to establish a web connection. This HTTP traffic will be intercepted and the client's web browser will be redirected to the captive portal web page.

The captive portal web page will instruct the client how to properly authenticate with the gateway machine. The client will then follow the presented authentication protocol, which may vary from client to client, based on its privilege level. The authentication information is passed between the client and the portal using SSL. The policies of the authentication system may permit some services to be provided to a "casual user".

Once the authentication is completed, the client sets up an encrypted tunnel between itself and the routing module of the gateway. Details about how this tunnel is implemented and set up are included in the next section. The routing module shares a separate encrypted tunnel with each authenticated client. The gateway allows only traffic originating through these encrypted tunnels to reach the Internet. All other wireless traffic cannot pass though the routing module. The routing module properly routes traffic between the subnet and the Internet. The subnet could be a private subnet with non-routable IP addresses, or it could use real Internet accessible IP addresses.

Note that this method eliminates the vulnerabilities of the 802.11b standard. The weak WEP protocol is not used. MAC filtering is not employed. Only clients having the key for its assigned encrypted tunnel are allowed to access the Internet. Since the clients do not share any encrypted tunnels; there is no possibility of data compromises, due to eavesdropping.

The routing module also provides a bandwidth shaping function. Due to the fact that the routing module has separate tunnels with each client; it is aware of the authenticity level of the client. Paying customers or regular users are provided with a high level of quality of service. Other casual or anonymous users are provided with limited bandwidth and with some other restrictions on port access to ensure that disruptive behavior is not facilitated. As stated before, all traffic not coming from tunnels is redirected to the captive portal and cannot reach the Internet.

4. IMPLEMENTATION

The generalized implementation overview, presented above, consisted of the following items: obtaining an IP address; blocking non-authenticated traffic; authenticating to the gateway; then, permitting the newly authenticated client traffic to pass; and traffic shaping.

One possible implementation of this model would be to use SSL for both the initial authentication and for providing the encrypted tunnel. SSL has had great success in authenticating web traffic and is available for many platforms and configurations. However, the SSL approach is not feasible; unless all the networking applications are reconfigured to use SSL; instead of regular sockets. Therefore, using SSL would violate our goal of minimal intrusiveness.

We used IPsec for creating the tunnel, as is done with popular implementations of the VPN mechanism. IPsec is available on a wide number of recent platforms, although it is, perhaps, overtly more complex than necessary [3]; it is best suited for this gateway. Given the desired system parameters and the client requirements, Windows 2000 was chosen as the client platform, while OpenBSD was chosen as the gateway platform.

Windows 2000 was the obvious choice for the client. It is a very popular business operating system that already has a built in IPsec implementation. Its popularity also makes it very likely to be in use by the users who would be interested in using the secure wireless gateway.

Even though Windows 2000 is the operating system used in the test implementation of this wireless gateway, it is not the only possible client operating system. Any OS, with a standards compliant IPsec client that uses ISAKMP may be used to connect to the secure wireless gateway. Many commercial IPsec clients are available for other Windows operating systems. In addition, there are many UNIX-like operating systems that have IPsec capabilities.

Due to OpenBSD's obscurity, our motivation for choosing OpenBSD as the gateway operating system may not be as obvious. OpenBSD is a freely available open source derivative of

the 4.4 BSD operating system. It is well known and respected for its reliability and security in a networked environment. Its security and integrated cryptography, including the *KeyNote Trust Management System* [4], make it an ideal choice.

Using IPsec provides the most versatile gateway solution. IPsec is designed to allow encrypted and authenticated network traffic between host machines over an existing TCP/IP network. IPsec and its associated authentication mechanism, ISAKMP will manage the authentication and encryption phases between the client and gateway computers. This combination will allow the administrator of a gateway to choose the best authentication mechanism for each case. Authentication could be based on preshared secret keys, or it could utilize a certificate based authentication scheme. This adaptability makes the gateway implementation appropriate for a large number of situations.

4.1 IPsec Background

IPsec is an attempt to provide additional security to IPv4 based networks, while maintaining compatibility with the existing network infrastructure. It provides two new network protocols: the Authentication Header (AH) [5] and the Encapsulating Security Payload (ESP) [6], which together can encrypt and authenticate network data packets.

IPsec has two modes of operation: the tunnel mode and the transport mode. The tunnel mode protects the original IP header and reveals only the IP address of the IPsec gateway machine. The transport mode does not protect this original IP header and encrypts only the payload.

In addition to the AH and ESP protocols, IPsec specifies a key management protocol: ISAKMP. ISAKMP negotiates which encryption and authentication algorithms are acceptable for use. It then handles the initial authentication and key exchange and all future key exchanges for the given session.

IPsec is an extremely flexible suite of protocols; thus, it is extremely complex. Much of this complexity is handled by ISAKMP, and in the end, we are provided with a versatile solution for negotiating connections and handling authentication. We utilize this flexibility by providing the owner of a gateway with options between authentication mechanisms. With the help of ISAKMP, the gateway administrator can choose either shared key or certificate based authentication.

4.2 Gateway Configuration

Our evaluation gateway machine is a Micron laptop with a 133MHz processor and 32 MB RAM, running OpenBSD 2.9. The wired network interface card is a Dlink DWE 650 PC-card. The client machine is a Compaq Laptop with a 600MHz processor, and 128 MB memory. The wireless network interface card, used by both the client and the gateway machine, is a Buffalo WLI PCM-11 PC-Card. We ran all the modules, captive portal, and routing on the single OpenBSD gateway machine.

The routing module is a simple, yet essential component. The OpenBSD Gateway must simply route all Internet destined packets from the wireless subnet to the appropriate network interface and vice versa. Our implementation uses a private range of IP addresses with the Gateway using Network Address Translation (NAT) to maintain stateful traffic information for our subnet. Routing and NAT are both closely related to packet filtering or firewalling. The user space programs ipf and ipnat are responsible for configuring this functionality on our gateway.

The proper configurations of the firewall and NAT modules are essential to ensure the functionality of the gateway. The firewall rules specify what to do with packets that reach a network interface, based on the packet's protocol, port number, or destination address.

On the wireless network interface the default rule is to drop all incoming packets, unless the packet is:

1. An ESP packet (tunneled IPsec traffic) from any IP address, destined for the wireless gateway.
2. From any IP address, destined for port 80 at any IP address
3. From any IP address, destined for port 500 on the wireless gateway
4. From any IP address, destined for port 53 on the DNS server specified by the DHCP lease

No traffic on the wireless network will be forwarded; unless it is IPsec traffic (ESP), as indicated by the first exception. Traffic on port 80 (HTTP traffic) is permitted, but will be handled by the NAT module and redirected to the web server on the gateway machine. This combination ensures that web traffic that has not passed through an IPsec tunnel will only reach the gateway machine, but never reach the Internet.

The third exception is necessary to allow ISAKMP to perform its security negotiations and key exchange. The final exception allows access to DNS service.

As indicated earlier, all HTTP traffic that did not pass through the IPsec tunnel will automatically be redirected to the web server on the gateway machine. Any client who is not using IPsec will only see the web page provided by the gateway machine. This process allows the gateway machine to provide instructions to clients, who have just connected to the network, but have not yet established an IPsec tunnel with the gateway via ISAKMP.

ISAKMP was chosen to handle the negotiation of IPsec tunnel connections and all session key exchanges. An initial, proof of concept, configuration establishes two different types of IPsec connections: public and private, using shared key authentication. The public connection will use a shared key that will be presented to any user who does not have a pre-established arrangement with the owner of the gateway. This key will be on the captive portal web page on the gateway machine, along with instructions on how to configure their clients.

Users, who have made arrangements with the owner of the gateway machine, will already have their shared key for authentication and instructions on how to use it.

To evaluate specific authentication and encryption algorithms for performance; we have restricted our gateway to use tunnel mode, with 3DES, as the encryption algorithm and HMAC_SHA, as the authentication algorithm. These choices are of reasonable security and are guaranteed to work in our cross platform environment.

4.3 Client Configuration

One of the major requirements of this project was to have a minimal impact on the end user, who will be using Windows 2000

(and can be extended to Windows XP). Therefore, the client configuration is very short and simple. The client machine must be configured to obtain its IP address automatically from a DHCP server. Once the user logs on and attempts to view any web page, the web browser displays the captive portal web page. At this point, the client is not authenticated with the IPsec gateway.

The captive portal web page provides the users with two options for configuring their client machines for use with the gateway. The first option walks the users through a few steps to configure the Windows 2000 IPsec subsystem. This uses the built in "IPsec Policies Management" snap-in in the Microsoft Management Console.

The second configuration option allows the users to download a Microsoft supplied tool (ipsecpol.exe) and a short script that will automatically configure the client machine. The ipsecpol.exe tool is the only way to configure the IPsec subsystem from the command line. The fact that the command line tool is not included with the operating system seems to merely be an oversight on Microsoft's part. This tool may already be installed on Windows 2000 machines that have the Windows 2000 Resource Kit installed.

The public users will receive their shared key with the configuration instructions on the SSL protected captive portal page. Since the private class users have a pre-established agreement with the owner of the gateway machine; the simplest method of configuration is using the ipsecpol.exe tool and a configuration script that contains the shared key.

Of course, some situations may require a more sophisticated and scalable authentication method. For those instances, certificate based authentication would be ideal and easily implemented. A certificate based solution makes more sense in an enterprise type setting, where a public key infrastructure is already in place. A haphazard PKI could result in a loosely bound certificate structure, which may be vulnerable to attacks.

5. PERFORMANCE

Performance of the above gateway configuration depends on the number of client machines, the particular IPsec algorithms utilized, and the speed of the gateway processor. Throughput measurements are summarized in Table 1. The tests performed involved the HTTP download of a 1 MB file between the client on the wireless network and a server on the wired network, local to the gateway machine. This test was done a few times to check the sanity of the tests, but extensive testing was not performed since there is adequate IPsec performance testing elsewhere.

The client machine was a COMPAQ laptop with a 600MHz processor and the gateway machine was a MICRON laptop with a133MHz processor.

Table 1 - Throughput

Unencrypted	604 KB/s
WEP (40bit)	458 KB/s
IPsec (DES/MD5)	355 KB/s
IPsec (3DES/SHA)	209 KB/s

It is clear that using IPsec degrades performance. The performance we obtained was limited due to the fact that encryption was done on a slow machine and a faster processor at the gateway would narrow the performance gap between the unencrypted and encrypted communications.

As can be clearly seen, the algorithm used can greatly influence the throughput. The gateway can specify which algorithm to use during the ISAKMP negotiation phase. It is also obvious that throughput will drop with the addition of extra clients to the network.

Despite all of these issues, the use of IPsec is still acceptable, given the intended function of the gateway. Since the gateway is intended to be an access point to the Internet; it stands to reason that the gateway will have a relatively low speed, less than local area network speeds, upstream network connection. It is likely to be able to handle traffic to several wireless clients, on average, at an acceptable speed.

In cases where a higher throughput is required three solutions exist. If the reduction in security would be acceptable, the gateway could be configured to use a less CPU intensive set of algorithms for IPsec. Another option would be to replace the rather underpowered 133 MHz gateway machine with a more substantial computer.

Perhaps the best solution for those who need the security and the additional throughput, would be to use a hardware cryptographic accelerator card. There are a number of crypto cards or network interface cards that handle DES or 3DES encryption in hardware. This could potentially bring throughput back up very close to its original throughput, without sacrificing security.

6. FUTURE WORK

Future work on this wireless gateway will improve configuration convenience, expand authentication options, and perhaps include facilities for bandwidth shaping.

An important feature of any system project should be ease of use by those who must interact with the system in any way. Handling gateway configuration via a simple script interface would greatly simplify the configuration process. Simple configuration scripts could be easily written, however with advanced authentication techniques the automation of the configuration process becomes more difficult.

Additional authentication options would add considerable value to this wireless gateway solution. Specifically, adding the ability to authenticate to the gateway, using a certificate-based solution, would be ideal. A certificate-based authentication scheme would be ideal for an enterprise, which already utilizes certificate-based authentication elsewhere.

The addition of bandwidth shaping functionality with the goal of providing guaranteed service to a class of users would bring this gateway to full maturity.

7. CONCLUSION

Our wireless gateway solution utilizes commodity operating systems and technologies to resolve fundamental security issues in an innovative new networking technology. The solution requires inexpensive gateway hardware and a proven operating platform. To maximize end user convenience, the solution is targeted to work with a common client operating system with no

client side modifications. The result is a secure, reliable, and easy to use wireless extension to any home, school, or corporate LAN with the intent of providing convenient Internet access to the user.

The single machine gateway solution can be tailored for specific operating environments. With appropriate hardware support, a sufficiently fast processor and hardware-accelerated cryptography, this gateway solution could easily be constructed on a simple to install and configure embedded gateway platform. This embedded gateway could easily replace today's standard, vulnerable wireless access points which are quickly being mapped, probed, and very likely abused by potentially malicious individuals.

8. REFERECES

[1] Fluhrer, S., Mantin, I. and Shamir, A. 2001. Weakness in the Key Scheduling Algorithm of RC4. *Eighth Annual Workshop on Selected Areas in Cryptography,* Toronto, Canada, August 2001.

[2] Stubblefield, A., Ioannidis, J., Rubin, A., "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP", ATT Labs Technical Report, TD4ZCPZZ, August 2001.

[3] Ferguson, N., and Schneier, B. "A Cryptographic Evaluation of IPsec." Preprint, January 2000.

[4] Blaze, M., Feigenbaum, J., and Keromytis, D. "The KeyNote Trust Management System." RFC 2704, September 1999.

[5] Kent, S., and Atkinson, R., "IP Authentication Header," RFC 2402, November 1998.

[6] Kent, S., and Atkinson, R., "IP Encapsulating Security Payload (ESP)," RFC 2402, November 1998.