Securing Wireless Networks in a University Environment

Jochen Eisinger, Peter Winterer, Bernd Becker
Institute of Computer Science
Albert-Ludwigs-University
D 79110 Freiburg im Breisgau, Germany
{eisinger,winterer,becker}@informatik.uni-freiburg.de

Abstract

Many applications in eLearning utilize wireless networks (WLAN) as a carrier for data and communication. These networks in their basic form are insecure, protecting the communication is a pre-condition for using a WLAN for confidential contents. Many existing solutions either do not provide the neccessary security or fail when employed with large user bases. The IPSec based solution developed at the Faculty of Applied Science of the University of Freiburg provides a high level of security, and by using digital X.509 certificates the problem of administrating a large user base is solved as well.

1. Introduction

Todays applications in eLearning depend on interaction between different peers. Contrary to classical communication where all peers are stationary, eLearning requires wireless communication including standards like Bluetooth [2] or the widely adopted 802.11 standard [8].

Solutions based on 802.11 are supported by a wide range of devices today, and provide a convenient way to access the network infrastructure. However, 802.11 based solutions introduce a large-scale security risk. Communication over such a wireless network is potentially insecure. All communication is unprotected and can easily be eavesdropped on or even spoofed.

This security problem was addressed during the development of the 802.11 standard. However, the so-called Wired Equivalent Privacy (WEP) protocol failed to provide the security its name promised [4]. The Wi-Fi Protected Access (WPA) [18] and its successor WPA2 (802.11i) [9] addressed this problem. Both standards rely on passwords for peer authentication. This and the fact that it is virtually impossible to enforce strict password policies in a University environment render these approaches unusable in our case as well. The encryption can only be as strong as the weakest link, in this case, the keying mechanism.

Another common approach to this problem is to use an ordinary tunneling protocol to protect the communication and authenticate the peers. Due to the nature of a wireless network, following issues have to be addressed: possible attackers may easily eavesdrop the communication; spoof the identity of the communication partner; or inject data in the communication at any time.

The most popular approach is to use the Point-to-Point Tunneling Protocol (PPTP) [7] to secure a WLAN. This protocol is easy to implement and configure. Moreover the PPTP server can use common protocols like LDAP or RADIUS for user authentication. It has been proven however that the PPTP user authentication is insecure and the encryption used is questionable [12, 13]; our own analysis showed that these weaknesses are exploitable [3].

IPSec [6] based solutions can provide a much higher level of security. IPSec however is a complex orchestration of various protocols. While it is possible to achieve secure communication using IPSec, it requires careful selection of implementation details and tuning of the various parameters. Popular IPSec based solutions often utilize a non-standard extension called XAUTH [10]. This extension uses passwords to authenticate peers. Similar to PPTP, the main advantage of this approach is the easy integration into existing user management systems. Despite its popularity, the XAUTH protocol was proven to be insecure, exposing the users password [14].

IPSec itself offers pre-shared keys or public key cryptography to authenticate peers. Both methods pose a key distribution problem. In the case of public key cryptography this problem can be solved using a public key infrastructure (PKI). The most common solution here are RSA keys with X.509 digital certificates [16].

Considering that IPSec implementations for all major operating systems (including handhelds) are available, it

should be straight forward to implement a secure wireless network. Still the implementation and configuration of such a network requires careful analysis. Otherwise, the resulting network may either be too heavily secured hindering communication, or the network does not provide the required level of security.

We will now present a setup developed for the Faculty of Applied Science of the University of Freiburg. This setup achieves a high level of security using IPSec and X.509 digitial certificates while being both easy to maintain and use. The setup includes: the setup of a 802.11 based wireless network; implementation of an IPSec security gateway; evaluation of various IPSec client software solutions; and the implementation of an X.509 based PKI. The respective parts of our setup will be presented in this order.

The PKI software developed for this purpose especially differs from other solutions in its high grade of integration into the whole wireless setup. As we will see later, this is one main reason, our solution can provide the flexibility of usage it does.

It turns out that with our solution maintaining a tightly secured wireless network is as easy as with comparable yet less secure solutions: The carefully tailored IPSec configuration avoids known security flaws and pitfalls; because of the PKI software developed for managing the RSA key pairs, the users can configure the IPSec client and use the wireless network without any insight into the IPSec protocol or the details of X.509 digital certificates.

2. IPSec Setup

There exists a wide range of solutions for securing communication over a network. We have decided to implement IPSec as security protocol. For one this protocol is believed to be secure, if applied correctly. The usage of IPSec for securing wireless communication is recommended by the German Bundesamt für Sicherheit im Internet (BSI) [5]. On the other hand, other available solutions like WEP, PPTP, or WPA(2) are suffering from security flaws or shortcomings of the authentication scheme mentioned before.

IPSec was first developed as part of the IPv6 standard and was later adopted for IPv4. IPSec was developed with support for a wide range of scenarios. Thus IPSec is not a single protocol, but a complex orchestration of different protocols that can be arranged to fit various needs. For our setup we have chosen the ESP protocol in tunnel mode, allowing a wide range of ciphers and message digest algorithms believed to be secure.

IPSec defines two protocols, the Authentication Headers (AH) and the Encapsulating Security Payload (ESP). The AH protocol offers data origin authentication, integrity of data, and anti-replay services. The ESP protocol provides data encryption and thus integrity of data, data origin authentication, and anti-replay services. Both protocols can be used either alone or together. For both protocols there is a tunnel mode and a transport mode.

The Transport mode modifies IP packets by adding the AH or ESP headers and transforming the data payload in the case of ESP. Here only parts of the IP header and the complete data payload are protected by IPSec. The transport mode is primarily meant for peer to peer communication.

The tunnel mode wraps the original IP packet in a new IP packet with the additional AH or ESP headers. In tunnel mode, the whole original IP packet is protected by IPSec. The tunnel mode was designed for gateway to gateway communication, protecting traffic between two networks.

We have choosen the ESP protocol in tunnel mode for several reasons. First of all, the tunnel mode is commonly used for connecting single clients to a larger network: the single client can be seen as a network with only one IP. The communication is then equivalent to a router to router setup. Second the ESP protocol offers a higher level of security when used in tunnel mode, since the whole IP packet is encrypted then. Otherwise, information about the destination address was visible on the wireless network. The AH protocol on its own does not provide encryption. On the other hand, the ESP protocol includes authentication of origin: if the packet is encrypted with the correct key, you can assume it was not fake. Using AH and ESP together does not add any value in our circumstances. This is meaningful only, when the authentication offered by ESP alone does not suffice.

We allow multiple encryption algorithms because this allows peers to choose an implementation especially optimized for their hardware. This is useful for handheld devices or other clients with limited computational power. Currently we are supporting the following encryption algorithms: 3DES, AES, BLOWFISH, TWOFISH, SERPENT, and CAST, and the following hash algorithms: MD5, SHA1, and SHA2. All those algorithms are currently believed to be secure. Of course this may change in the future. For example the MD5 hash algorithm appears to suffer from design flaws and may have to be removed in the future [17].

We have implemented the IPSec configuration described above and put to use at the University of Freiburg [1]. On the server side we are using a Debian GNU/Linux installation with FreeS/WAN as the IPSec software. FreeS/WAN was patched with X.509 support and support for the additional ciphers and message digests. The stock FreeS/WAN software only supports 3DES. Meanwhile these patches are included in the two spin-offs StrongS/WAN and OpenS/WAN. FreeS/WAN itself was discontinued after the release of the Linux kernel version 2.6.

The Linux based IPSec gateway was in use for about one and a half year, when we introduced a second hardware based IPSec gateway. We are deploying a Cisco VPN Concentrator 3060 as an IPSec gateway. While the hardware accelerated IPSec gateway is considerably faster the configuration of the Cisco VPN Concentrator is more complex due to the waste number of options available.

3. Client-side Support

On the client side various solutions are used. Operating systems with native IPSec support include Microsoft Windows 2000 and XP, Linux kernel version 2.6, FreeBSD, and Mac OS X. Other systems that are known to have native IPSec support but not tested include NetBSD and OpenBSD. Those operating systems can use both the Linux and the Cisco IPSec gateway without any third party software.

The Cisco VPN client which is mainly intended for connecting to the Cisco VPN Concentrator is available for Microsoft Windows, Linux and Mac OS. For Linux the FreeS/WAN spin-offs StrongS/WAN and OpenS/WAN also exist.

For Palm and PocketPC devices we have evaluated the MovienVPN client. This client however does not support RSA key authentication due to the restricted computational power of those devices. Instead, it supports the insecure XAUTH protocol or pre-shared key authentication. For certain handhelds, like the Sharp Zaurus, there are open source based operating systems available. The Linux based Opie for example can be customized with FreeS/WAN in order to provide native IPSec support. For other handhelds, a tradeoff between security and speed has to be made. Current handhelds however already provide enough computational power to use RSA based authentication schemes. It is to be expected that more IPSec clients with support for RSA and X.509 digitial certificates will appear for handheld devices.

When comparing native IPSec support to third party tools like the Cisco VPN dialer, two things can be observed. Native support of course is less intrusive, because the operating system does not need to be altered. A third party IPSec client has to alter the network stack of the operating system and thus possibly conflicts with other software like personal firewalls. The downside of native IPSec clients is their configuration. Often different versions of the same operating system require totally different configurations for the IPSec client. On the other hand, a third party client supporting different operating system has similar configuration options on all supported operating systems. Such a client is easier to maintain and support.

4. Authentication

While our choice of encryption algorithms allows for secure communication, the issue of authentication is yet to be discussed. No matter how strong the encryption used for communication is, if the authentication is weak and a possible attacker can gain access to keying material or, even worse, to credentials, the whole system is flawed.

The standard authentication options of IPSec include pre-shared keys (PSK), RSA based public key authentication and X.509 digital certificate based authentication. A PSK differs from a normal password such that both the client and the server know the plain password, whereas normally only the client has the plain password and the server only has a hashed version. Besides this subtle difference, a PSK can be seen as a normal password. However, due to this difference, normal password sources cannot easily be used, since they only hold the hashed password, while IPSec would need the plain password in order to use PSK for authentication. A possible solution to this problem would be to use the password hash as PSK. However, this would restrict the length of the PSK to the length of the hash and thus weaken the authentication system.

RSA based authentication on the contrary, requires only the peers to know each others public key. This is advantageous because less sensible information has to be stored. Furthermore, public key based authentication is believed to be the most secure authentication scheme available for IPSec [15]. Since the server needs to know every users public key, a huge administrative problem remains. This problem is commonly solved by a Public Key Infrastructure (PKI).

For our wireless network, we have decided to implement X.509 digital certificate based authentication. For the X.509 based authentication to work we need a PKI. A PKI helps to identify valid public keys. A central certification authority (CA) digitally signs certificates, including information about the owner of the public key and the key itself. With such a certificate, it is always possible to prove that a given public key belongs to a certain individual. When using digitial certificates with IPSec, the peers exchange public keys and certificates prior to the authentication phase. If the certificates are valid, the certified owner is allowed to establish a connection, and the authentication phase starts. Exchanging the certificates is not part of the authentication since the certificates as well as the public keys do not contain any confidential information. The authentication is done later using the private keys.

Assuming the user already has a RSA key pair and a certificate, installing the IPSec client and importing the key pair is enough to connect to the network. The server configuration is even more simple than compared to XAUTH,

since the server does not need to query an authentication server but uses the information provided in the digital certificate.

There exist several PKI solutions, both commercial and opensource. Most of these implementation are quite feature rich, which at first would seem advantageous as it allows you to customize and optimize your network. However, since most of the options of a full featured PKI solution are not needed, the setup and usage can be overly complex. For example the creation of a new certificate usually requires filling out a form with the users name, organization and other data. As simple as this may appear, this is a source of possible errors and will lead to problems when several hundred or thousand of users go through this every term. Another problem is the duration of the process. The data the user has entered needs to be verified before the certificate is signed. Some solutions even require the user to create a RSA key pair. While all these steps are neccessary to create a digitial certificate, most of these steps can be automated.

This is why we have decided to implement a proprietary PKI software [11] for our project. We have based our implementation on the OpenSSL software package and added a web frontend using the web scripting language PHP. OpenSSL is a highly configurable cryptographic software package that includes support for the X.509 protocol. OpenSSL is well documented and widely used free software, however its usage is too complex for an ordinary student. With the help of our web interface, managing certificates does not require any knowledge of OpenSSL or certificates. The web interface also connects the user management database to the OpenSSL software package.

Figure 1. Certificate management dialog

A user that wants to create a certificate can log on to the web interface with his or her credentials. Depending on whether a certificate already exists and the user account is enabled for wireless access, the user may either create, download, or delete an existing certificate. Figure 1 shows the dialog where a user can create such a certificate. The certificate will be stored on the server, so the user can download it again. To protect the certificate, the PKCS#12 file is secured by a password choosen by the user.

2002											
								15	272	101	27

2003											
48	15	18	187	109	42	21	11	24	230	75	27

2004											
50	20	38	198	98	50	25	19	14	150	113	21

Table 1. New certificates created per month

5. Evaluation and Discussion

We have evaluated this solution in the Faculty of Applied Science campus at the University of Freiburg. At the time of writing, approximately 1500 students can potentially use this implementation to access the wireless network. The wireless network currently spans the whole Faculty of Applied Science and selected spots around the rest of the University. Approximately 200 students are currently using the wireless network. We allow students to create certificates on a per term basis. Accordingly, the highest number of new certificates created is at the beginning of each term (see table 1). Note that commonly more certificates are created than certificates being used. Investigating that issue showed that students either just want to try out the functionality of the web interface or simply forgot the password they had chosen to protect the certificate.

Usage statistics show that during lecture periods up to 60 students are using the wireless network at the same time, while during the term breaks, at most 10 students are online at the same time (see figure 2, the graph shows the number of active IPSec sessions on the Linux gateway). Towards the end of figure 2, the number of sessions is decreasing because of the introduction of the Cisco VPN Concentrator as alternative IPSec gateway. Usage statistics are not yet available for the new gateway.

During the two years our implementation was used, the acceptance by the students was very high. Still some complain about an overly complex setup procedure. We are commonly faced with two different arguments. Some do not see the necessity of securing wireless communication at all. For an individual this might be true, if somebody is only using the wireless network to read news sites, there is

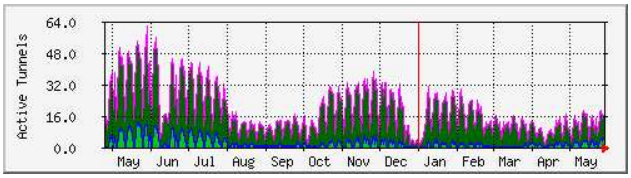

Figure 2. active IPSec sessions 2003/2004

no need to secure this communication. However when using the wireless network for more serious tasks like working on research topics or other confidential data, encryption is mandatory. The other complaint is about the overall procedure being too complex. Of course just plugging in a wireless card and maybe entering a WEP key is easier then installing additional a software. However, the software installation has to be done only once. And as outlined above, the only easier solutions, like no encryption at all or WEP, do not provide any security. In this respect, the argument boils down to the same point as the first argument, why is there the need for any security at all.

It is not obvious to everybody using the wireless network that encryption is a must. Yet nobody so far has voluntarily revealed his or her mail account password when insisting on encryption being superfluous.

6. Conclusion

Often wireless networks are not protected or are protected using insecure solutions, which is often worse because it lulls the users in a false feeling of security. This lack of security is commonly justified by the overly complex and labour intensive configuration and administration of appropriate security measures.

Contrary to that, we have shown that it is possible to maintain a high level of security while keeping the administrative efforts equally low as with other solutions. As for acceptance by the users the experience of over two years has shown that the setup and usage of our approach is not more difficult than other solutions. In fact the wireless network is used by students and staff members on a regular basis and thus provides a solid foundation for eLearning applications.

Acknowledgements

The authors appreciate the helpful discussions with Ilia Polian from University Freiburg, Germany.

References

[1] B. Becker. *Mobile Pools*. Institute for Computer Science, University of Freiburg. `http://mopoinfo.vpn.uni-freiburg.de/mopo-projekt.php`.

[2] The Bluetooth Special Intereset Group (SIG). *Specification of the bluetooth system*, version 1.1 edition, 2001. `http://www.bluetooth.com`.

[3] J. Eisinger. *Exploiting known security holes in Microsoft's PPTP Authentication Extensions (MS-CHAPv2)*, 2001. `http://mopoinfo.vpn.uni-freiburg.de/pptp_mschapv2/`.

[4] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of rc4. In *Lecture Notes in Computer Science*, volume 2259, pages 1–24, 2001.

[5] Bundesamt für Sicherheit in der Informationstechnik. *BSI-Broschüre: Sicherheit im Funk-LAN (WLAN, IEEE 802.11)*. `http://www.bsi.bund.de/literat/doc/wlan/index.htm`.

[6] The IPSEC Working Group. Ip security protocol (ipsec) charter. `http://www.ietf.org/html.charters/ipsec-charter.html`.

[7] The Network Working Group. Point-to-point tunneling protocol (pptp), 1999. RFC 2637.

[8] IEEE Std 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification*, 1999 edition edition.

[9] IEEE Std 802.11i. *Medium Access Control (MAC) Security Enhancements*, 2004 edition edition.

[10] R. Pereira and S. Beaulieu. *Extended Authentication within ISAKMP/Oakley*, 1999. draft-ietf-ipsec-isakmp-xauth-04.txt.

[11] The Mobile Pools project (MoPo). *Certificate Management Software*. `http://mopoinfo.vpn.uni-freiburg.de/ca/software`.

[12] B. Schneier and Mudge. Cryptoanalysis of microsoft's point-to-point tunneling protocol (pptp). In *Proceedings of the 5th ACM Conference on Communications and Computer Security*, pages 132–141. ACM Press, 1998.

[13] B. Schneier, Mudge, and D. Wagner. Cryptoanalysis of microsoft's pptp authentication extensions (ms-chapv2). In *CQRE'99*, pages 192–203. Springer Verlag, 1999.

[14] T. L. Simon. *Multiple vulnerabilities in vendor IKE implementations, including Cisco*, 2003. `http://www.securityfocus.com/archive/1/347351`.

[15] A. Steffen. Virtual private networks coping with complexity. In *Lecture Notes in Informatics*, volume P-44, pages 289–302. Bonner Köllen Verlag, 2003.

[16] International Telecommunication Union. *Recommendation X.509: Public-key and attribute certificate frameworks*, 2000. `http://www.itu.org`.

[17] X. Wang, D. Feng, X. Lai, and H. Yu. Collisions for hash functions md4, md5, haval-128 and ripemd. In *Cryptology ePrint Archive*, volume 2004/199, 2004.

[18] Wi-Fi Alliance. *WPA Specification Documentation*, version 3.1 edition, 2003. `http://wi-fi.org`.