PROJECT REPORT

SECURITY IN WIRELESS NETWORKS

SUBMITTED IN PARTIAL FULFILMENT OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

by

Kulkarni Chaitanya Deepak (Y1-315, Sl.No.27)
Pallav Gupta (Y1-232, Sl.No. 36)

Under the guidance of

Dr. M.P. Sebastian

Department of Computer Engineering
National Institute of Technology, Calicut

National Institute of Technology, Calicut
Department of Computer Engineering

Certified that this Project Report entitled

SECURITY IN WIRELESS NETWORKS

is a bonafide report of the project presented by

Kulkarni Chaitanya Deepak (Y1-315, Sl.No.27)
Pallav Gupta (Y1-232, Sl.No. 36)

in partial fulfilment of the degree of
Bachelor of Technology
under our guidance

Dr. M.P. Sebastian
Asst. Professor
Dept.of Computer Engineering

Dr. V.K. Govindan
Professor and
Head
Dept.of Computer Engineering

Acknowledgement

We wish to make use of this opportunity to express our sincere feelings of gratitude to our guide **Dr. M.P. Sebastian**, Asst. Professor, Department of Computer Science and Engineering, National Institute of Technology, Calicut, for his invaluable guidance, co-operation, constant encouragement and advices. During the process of our work we have incurred obligations to many and we are greatly indebted to each and everyone of them. We thank them all for always being there for us.

Kulkarni Chaitanya Deepak (Y1-315, Sl.No.27)
Pallav Gupta (Y1-232, Sl.No. 36)

Abstract

A plethora of security protocol implementations, for wireless networks, exist today. The various mechanisms and algorithms that are webbed together into current suite of wireless security protocols have serious flaws. Enhancements to the existing protocols in the domain of Wireless Networks are severely needed.

Here we analyze the security aspects of existing authentication frameworks for wireless networks, namely 802.11 and 802.1x. The Authentication methods stipulated by 802.11, i.e. Use of SSID, Open Authentication, Shared Key Authentication and Client MAC Verification have their share of vulnerabilities. Although 802.1x supports port based authentication and has key management features, it suffers from lack of Mutual Authentication and Session Hijacking attacks. One way authentication is simply not enough because as only server authenticates the client, the client cannot be sure of the server's identity.

We then propose and explain our authentication and security system. We also present a study of RADIUS and EAP Protocols. RADIUS protocol for Authenticator Server and IEEE 802.1x protocol with EAP-TTLS have emerged as our protocols of choice after a comparative study of various protocols mentioned above. RADIUS follows a Client-Server Model and provides network security by the use of a shared secret which is never sent over the network and encryption. EAP is an Authentication Protocol. Its variant EAP-TTLS uses digital certificates and tunnelling for User Authentication. Our Mechanism makes use of these protocols and thus provides a robust security solution with strong mutual authentication.

Contents

1 Introduction

Last few years have seen a large number of Wireless Local Area Networks (WLANS) based on IEEE 802.11 protocols being deployed in a variety of places including homes, offices, colleges, airports etc. WLANS provide unethered connectivity to portable devices, such as laptops and PDAs. In some cases, WLANs can also serve as *last mile* connectivity technology. However, further widespred deployment of WLANs depends upon their security aspect.Network Managers have to provide end users with freedom and mobility without offering intruders access to information sent and received over the wireless networks. With a WLAN, transmitted data is broadcast over the air using radio waves. This means, any WLAN client within an Access Point (AP) service area can receive can receive data transmitted from or to the AP. Thus, if stringent security measures are not in place, installing WLANs can be considered equivalent to providing Ethernet ports everywhere, including parking lots.[1]

A host of Security and Authetication features exist in present WLAN protocols, not one of which is without its share of vulnerabilities. To mitigate various threats to WLANs, the network managers have to apply several layers of defence across the network. The Authentication Mechanism presented in this report forms only a part of the overall security system. Other components like Firewalls, intrusion detection systems and segmented networks must also be considered by network managers.

In this project report, we study security and authentication aspects of various existing WLAN protocols and propose our design of a Authentication and Security mechanism based on a combination of various protocols and mutual authentication. Following three sections describe IEEE 802.11, IEEE 802.1x and PPP and its optional authentication phase. We then move on to our mechanism.

1.1 Problem Definition

Study various existing Authentication and Security Protocols in the domain of Wireless Networks and analyze their vulnerabilities. Also provide a tightly coupled Authentication and Security Mechanism for authenticating mobile clients which try to connect to a access point and maintain a secure connection with them.

1.2 Scope of Our Work

We have studied various existing protocols, namely IEEE 802.11, IEEE 802.1x, RADIUS and EAP. Also, we have designed and implemented the authentication part of network communication protocol. We have not handled issues like seamless handling over of a client to a new access point from the previous access point. We have not dealt with WEP key management issues and per packet encryption as they are not part of authentication mechanism. The mechanism developed by us is tightly coupled, as it user based rather than device based.

2 IEEE 802.11 and its vulnerabilities

Wireless LANs, because of their broadcast nature, require the addition of:

- User authentication to prevent unauthorized access to network resources
- Data privacy to protect the integrity and privacy of transmitted data

The 802.11 specification stipulates two mechanisms for authenticating wireless LAN clients: **open authentication** and **shared key authentication**. Two other mechanisms **the Service Set Identifier (SSID)** and **authentication by client Media Access Control (MAC) address** are also commonly used. This section explains each approach and its weaknesses.

2.1 Service Set IDentifier

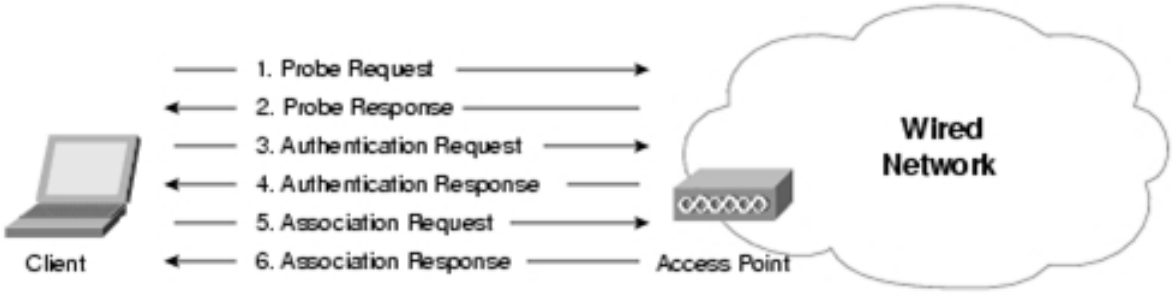
SSID is a 32-character unique identifier attached to the header of packets sent over a WLAN that acts as a password when a mobile device tries to connect to the BSS. The SSID differentiates one WLAN from another, so all access points and all devices attempting to connect to a specific WLAN must use the same SSID. A device will not be permitted to join the BSS unless it can provide the unique SSID. Because an SSID can be sniffed in plain text from a packet it does not supply any security to the network.

The SSID does not provide any data-privacy functions, nor does it truly authenticate the client to the access point.[2]

2.2 802.11 Station Authentication

Authentication in the 802.11 specification is based on authenticating a wireless station or device instead of authenticating a user. The specification provides for two modes of authentication: open authentication and shared key authentication.

The 802.11 client authentication process consists of the following transactions :



- Client broadcasts a probe request frame on every channel
- Access points within range respond with a probe response frame
- The client decides which access point (AP) is the best for access and sends an authentication request

- The access point will send an authentication reply
- Upon successful authentication, the client will send an association request frame to the access point
- The access point will reply with an association response
- The client is now able to pass traffic to the access point[2]

The next four subsections will detail individual client authentication processes.

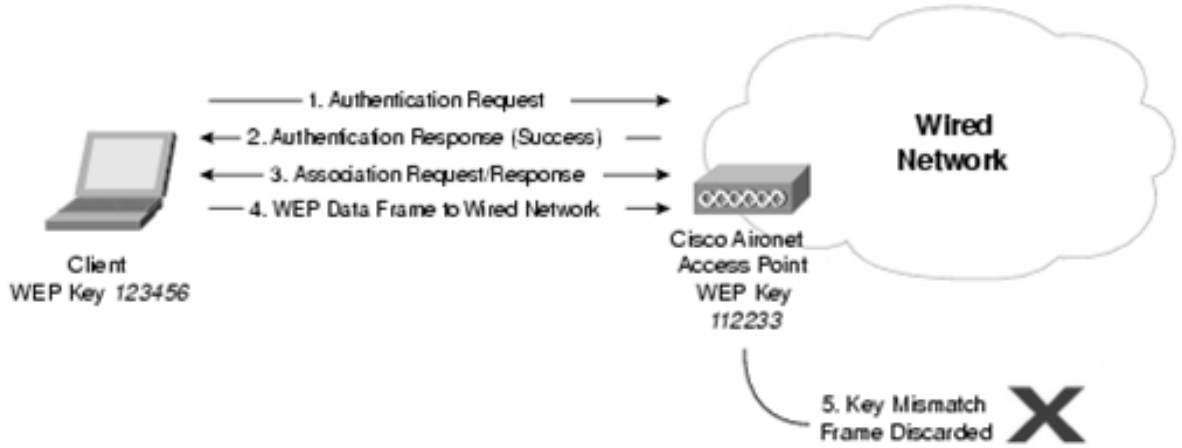
2.2.1 Probe Request and Response

Once the client becomes active on the medium, it searches for access points in radio range using the 802.11 management frames known as probe request frames. The probe request frame is sent on every channel the client supports in an attempt to find all access points in range that match the SSID and client-requested data rates.[2]

All access points that are in range and match the probe request criteria will respond with a probe response frame containing synchronization information and access point load. The client can determine which access point to associate to by weighing the supported data rates and access point load. Once the client determines the optimal access point to connect to, it moves to the authentication phase of 802.11 network access.[2]

2.2.2 Open Authentication

Open authentication is a null authentication algorithm. The access point will grant any request for authentication. It might sound pointless to use such an algorithm, but open authentication has its place in 802.11 network authentication. Authentication in the 1997 802.11 specification is connectivity-oriented. The requirements for authentication are designed to allow devices to gain quick access to the network. In addition, many 802.11-compliant devices are hand-held data-acquisition units like bar code readers. They do not have the CPU capabilities required for complex authentication algorithms.



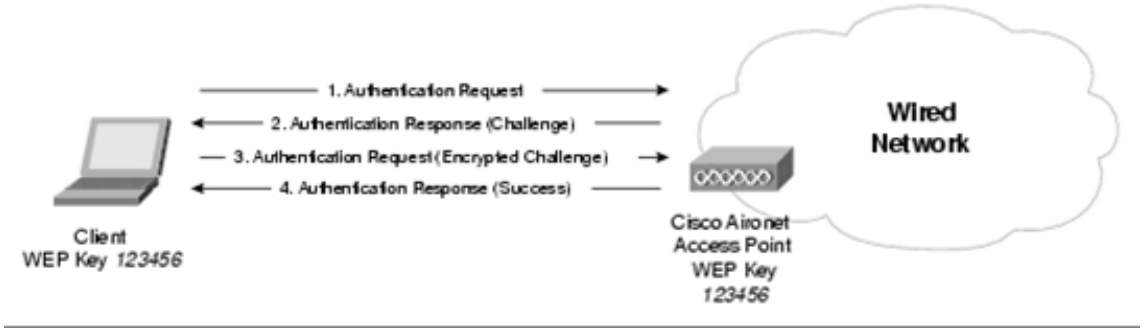
Open authentication consists of two messages:

- The authentication request
- The authentication response

Open authentication allows any device network access. If no encryption is enabled on the network, any device that knows the SSID of the access point can gain access to the network. With WEP encryption enabled on an access point, the WEP key itself becomes a means of access control. If a device does not have the correct WEP key, even though authentication is successful, the device will be unable to transmit data through the access point. Neither can it decrypt data sent from the access point.[2]

2.2.3 Shared Key Authetication

Shared key authentication is the second mode of authentication specified in the 802.11 standard. Shared key authentication requires that the client configure a static WEP key. Following figure describes the shared key authentication process.



- The client sends an authentication request to the access point requesting shared key authentication
- The access point responds with an authentication response containing challenge text
- The client uses its locally configured WEP key to encrypt the challenge text and reply with a subsequent authentication request
- If the access point can decrypt the authentication request and retrieve the original challenge text, then it responds with an authentication response that grants the client access.[2]

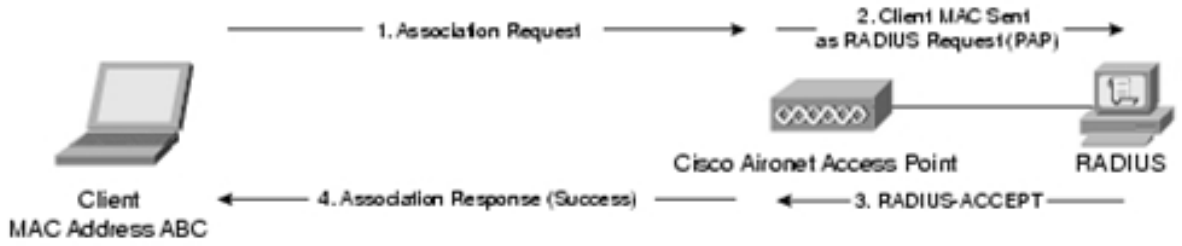
2.2.4 MAC Address Authentication

MAC address authentication is not specified in the 802.11 standard, but many vendors-including Cisco-support it. MAC address authentication verifies the client's MAC address against a locally configured list of allowed addresses or against an external authentication server. MAC authentication is used to augment the open and shared key authentications provided by 802.11, further reducing the likelihood of unauthorized devices accessing the network.[2]

2.3 Authentication Vulnerabilities

2.3.1 Use of SSID

The SSID is advertised in plain-text in the access point beacon messages. Although beacon messages are transparent to users, an eavesdropper can easily determine the SSID with the use of an 802.11 wireless LAN packet analyzer, like Sniffer Pro. Some access-point vendors,



including Cisco, offer the option to disable SSID broadcasts in the beacon messages. The SSID can still be determined by sniffing the probe response frames from an access point.

The SSID is not designed, nor intended for use, as a security mechanism. In addition, disabling SSID broadcasts might have adverse effects on Wi-Fi interoperability for mixed-client deployments.

2.3.2 Open Authentication Vulnerabilities

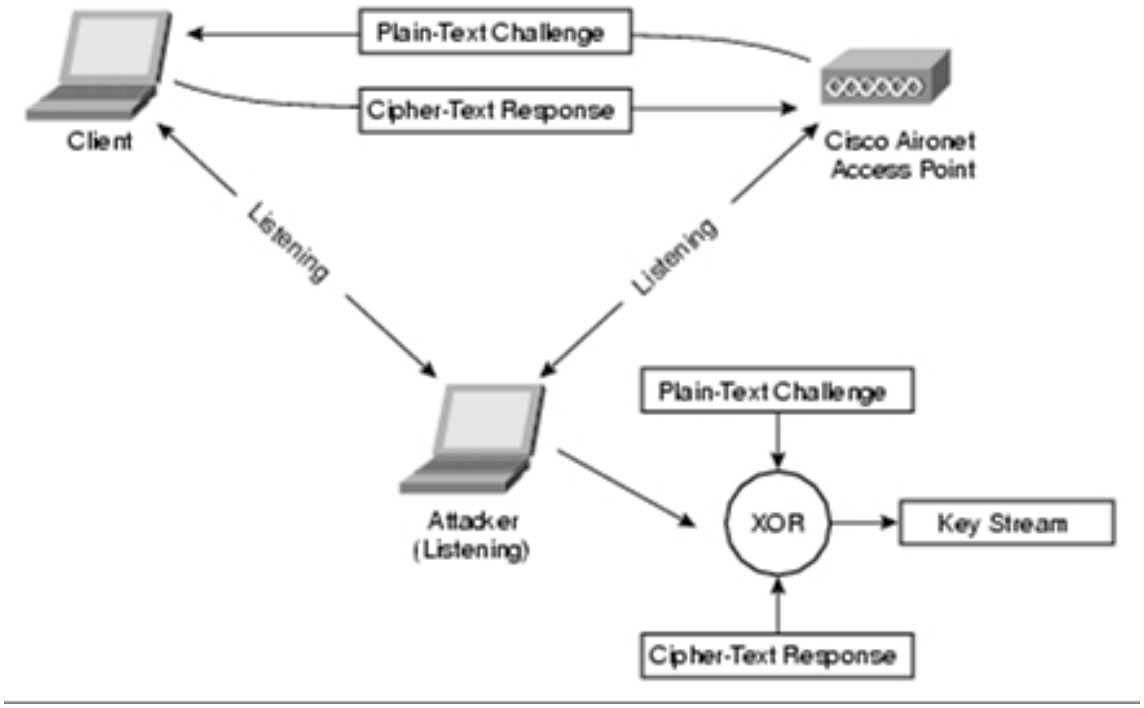
Open authentication provides no way for the access point to determine whether a client is valid. This is a major security vulnerability if WEP encryption is not implemented in a wireless LAN. In scenarios in which WEP encryption is not needed or is not feasible to deploy, such as public wireless LAN deployments; strong, higher-layer authentication can be provided by implementing a Service Selection Gateway (SSG).

2.3.3 MAC Address Authentication Vulnerabilities

MAC addresses are sent in the clear as required by the 802.11 specification. As a result, in wireless LANs that use MAC authentication, a network attacker might be able to subvert the MAC authentication process by "spoofing" a valid MAC address. MAC address spoofing is possible in 802.11 network interface cards (NICs) that allow the universally administered address (UAA) to be overwritten with a locally administered address (LAA). A network attacker can use a protocol analyzer to determine a valid MAC address in the business support system (BSS) and an LAA-compliant NIC with which to spoof the valid MAC address.

2.3.4 Shared Key Authentication Vulnerabilities

Shared key authentication requires the client use a preshared WEP key to encrypt challenge text sent from the access point. The access point authenticates the client by decrypting the shared key response and validating that the challenge text is the same.



The process of exchanging the challenge text occurs over the wireless link and is vulnerable to a man-in-the-middle attack. An eavesdropper can capture both the plain-text challenge text and the cipher-text response. WEP encryption is done by performing an exclusive OR (XOR) function on the plain-text with the key stream to produce the cipher-text. It is important to note that if the XOR function is performed on the plain-text and cipher-text are XORed, the result is the key stream. Therefore, an eavesdropper can easily derive the key stream just by sniffing the shared key authentication process with a protocol analyzer.

2.4 Wired Equivalent Privacy

Wired Equivalent Privacy (WEP) is part of the IEEE 802.11 standard (ratified in September 1999), and is a scheme used to secure wireless networks (WiFi). Because a wireless network broadcasts messages using radio, it is particularly susceptible to eavesdropping; WEP was designed to provide comparable confidentiality to a traditional wired network, hence the name. However, several serious weaknesses were identified by cryptographers, and WEP was superseded by Wi-Fi Protected Access (WPA) in 2003, and then by the full IEEE 802.11i standard (also known as WPA2) in 2004. Despite the inherent weaknesses, WEP provides a bare minimal level of security that can deter casual snooping.

WEP uses the stream cipher RC4 for confidentiality and the CRC-32 checksum for integrity. For RC4, WEP uses two key sizes: 40 bit and 104-bit; to each is added a 24-bit initialization vector (IV) which is transmitted in the clear.

2.4.1 WEP Flaws

Two generic weaknesses of WEP are:

- the use of WEP was optional, resulting in many installations never even activating it, and
- WEP did not include a key management protocol, relying instead on a single shared key amongst users.

More specific attacks have also become evident: in August 2001, Fluhrer et al. published a cryptanalysis of WEP that exploits the way the RC4 cipher is used, resulting in a passive attack that can recover the RC4 key after eavesdropping on the network for a few hours; the attack was soon implemented, and automated tools have since been released. It is possible to perform the attack with a personal computer, off-the-shelf hardware and freely-available software. Cam-Winget et al. write, "Experiments in the field indicate that, with proper equipment, it is practical to eavesdrop on WEP-protected networks from distances of a mile or more from the target."

In 2005, a group from the U.S. Federal Bureau of Investigation gave a demonstration where they broke a WEP-protected network in 3 minutes using publicly available tools.[5]

3 IEEE 802.1x and Its Vulnerabilities

IEEE 802.1x is a port-based authentication protocol. There are three different types of entities in a typical 802.1x network, including a supplicant, an authenticator and an authentication server. To permit the EAP traffic before the authentication succeeds, a dual-port model is used in IEEE 802.1x specifications. In an unauthorized (uncontrolled) state, the port allows only DHCP and EAP traffic to pass through.

3.1 Main Features of IEEE 802.1x

When applied to 802.11b, the 802.1x specification includes two main features:

- logical ports and
- key management.

3.1.1 Logical Ports

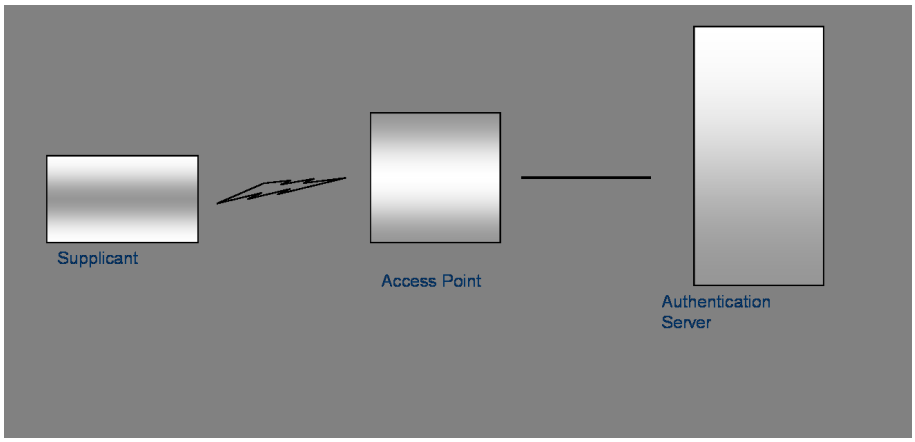
Unlike wired networks, wireless stations are not connected to the network by physical means. They must have some sort of association relation with an AP in order to use the WLAN. This association is established by allowing the clients and the AP to know each other's MAC address. This combination of MAC address of the AP and that of the station acts as a logical port. This then acts as a destination address in EAPOL protocol exchanges.

3.1.2 Key Management

IEEE 802.1x specifications do not emphasize on using WEP key for encryption. This is because key information is passed from an AP to a station using EAPOL-Key message. Keys are generated dynamically, at a per-session basis.

3.2 WLAN Configuration Using 802.1x

Typical configuration of a WLAN using IEEE 802.1x[7] is shown in Figure:



Access Point and Authenticator are physically connected by a wire. Wireless connection

exists between Supplicant and Access Point. Messages are exchanged between the Supplicant and the Authenticator to establish the Supplicant's identity. The Authenticator then transfers the Supplicant's information to the Authentication Server using RADIUS. All communications between the Authentication Server and the Supplicant passes through the Authenticator using EAP over LAN (i.e., EAPOL) and EAP over RADIUS, respectively. This creates an end-to-end EAP conversation between the Supplicant and the Authentication Server.

3.3 Vulnerabilities

In their celebrated paper, Arunesh Mishra and William Arbugh describe

- Lack of Mutual Authentication
- Session Hijacking

as the major vulnerabilities[3] of 802.1x.

3.3.1 Lack of Mutual Authentication

According to 802.1x specifications, a Supplicant always trusts the Authenticator but not vice versa. Consider Figure 2. There is no EAP Request message originating from the Supplicant (the client). It only responds to the requests sent by the Authenticator (the AP). This one-way authentication opens the door for "MAN IN THE MIDDLE ATTACK".

A man-in-the-middle is an approach typically used to be able to read a public-key encrypted conversation. It relies on having complete access to all messages between the two parties wanting to communicate A and B. Meaning all messages between A and B must pass between the man in the middle M. Upon the start of communication the public keys must be exchanges between A and B. This is where M starts to interfere by creating an own key-pairs for both A and B. They are distributed back to A and B in a way that M are able to decrypt, read and encrypt messages passing by. A and B will think they are communicating though a secure channel, but only the channel between A and M, and M and B is actually secured and M can read and modify all of their messages.

The EAP-Success message sent from the Authenticator to the Supplicant contains no integrity preserving information. An attacker can forge this packet to start the attack.

3.3.2 Session Hijacking

With IEEE 802.1x, RSN (Robust Security Network) association has to take place before any higher layer authentication. Thus we have two state machines. One is classic 802.11 and the other is 802.1x based RSN state machine. Their combined action should dictate the state of authentication. However, due to a lack of clear communication between these two state machines and message authenticity, "Session Hijacking Attack" becomes possible.

Session hijacking is the act of taking control of a user session after successfully obtaining or generating an authentication session ID. Session hijacking involves an attacker using captured, brute forced or reverse-engineered session IDs to seize control of a legitimate

user's Web application session while that session is still in progress.

There are three primary techniques for hijacking sessions:

- **Brute force:** the attacker tries multiple IDs until successful.
- **Calculate:** in many cases, IDs are generated in a non-random manner and can be calculated.
- **Steal:** using different types of techniques, the attacker can acquire the Session ID.

4 Our Solution : Protocols

Our solution to the abovementioned problem is the use of a combnination of protocols which impart Mutual Authentication to the Authentication and Security Mechanism designed by us.

4.1 Mutual Authentication

Authentication in an information system takes place in a client/server context, in which the individual user is the client and some computer is a form of server. A user is required to authenticate his or her identity to a server, usually as a prerequisite for gaining access to resources (access control or authorization). This is typically an explicit one-way authentication process; that is, the user authenticates himself or herself to the server. If the user is authenticating to a computer directly (for example, when sitting at a desktop or laptop computer), there is an implicit two-way authentication; the user sees the computer with which he or she is interacting and presumably knows that it is the one he or she wishes to use.

However, if the user is authenticating to a computer accessed via a communication network, there is often no way to verify that the computer at the other end of the communication path is the one that the user is trying to contact. The user typically relies on the communication infrastructure operating properly and thus connecting him or her to the intended computer. This assumption may be violated by any of a number of attacks against the communication path, starting with the computer that the user is employing locally. This lack of explicit, secure, two-way authentication can subvert many types of individual authentication mechanisms. If a user provides an secret(for eg. Password) to the wrong Authenticator, both security and privacy are adversely affected. Thus, two-way authentication is preferred so that an user can verify the identity of the Authenticating Server to which a secret may be disclosed.

As we have seen, Mutual Authentication is not a standard feature both with 802.1x and CHAP. One way Authentication creates possibilities of Session Hijack and Man in the Middle attack. One Way Authentication is clearly not safe and not enough.

We use RADIUS (Remote Access Dial In User Service) protocol to setup our Authentication Server. We provide a tightly coupled Authentication mechanism by providing User Authentication rather than Device Authentication. The shortcomings of WEP are overcome by using EAP-TTLS along with 802.1x. By providing Mutual Authentication, we mitigate the problems associated with one-way authentication.

We now discuss RADIUS protocol and EAP protocol.

4.2 RADIUS Protocol

RADIUS stands for Remote Authentication Dial-In User Service. RADIUS is a widely deployed protocol enabling centralized authentication, authorization, and accounting for network access. Originally developed for dial-up remote access, RADIUS is now supported by virtual private network (VPN) servers, wireless access points, authenticating Ethernet switches, Digital Subscriber Line (DSL) access, and other network access types.

4.2.1 Key Features of RADIUS

Following are the key features of RADIUS[8]:

- **Client/Server Model:** A Network Access Server (NAS) operates as a client of RADIUS. The client is responsible for passing user information to designated RADIUS servers, and then acting on the response which is returned. RADIUS servers are responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user.A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.
- **Network Security:** Transactions between the client and RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and RADIUS server, to eliminate the possibility that someone snooping on an unsecured network could determine a user's password.
- **Flexible Authentication Mechanisms:** The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the user name and original password given by the user, it can support PPP PAP or CHAP, UNIX login, and other authentication mechanisms.
- **Extensible Protocol:** All transactions are comprised of variable length Attribute-Length-Value 3-tuples. New attribute values can be added without disturbing existing implementations of the protocol.

4.2.2 RADIUS Operation

When a client is configured to use RADIUS, any user of the client presents authentication information to the client. This might be with a customizable login prompt, where the user is expected to enter their username and password. Alternatively, the user might use a link framing protocol such as the Point-to-Point Protocol (PPP), which has authentication packets which carry this information.

Once the client has obtained such information, it may choose to authenticate using RADIUS. To do so, the client creates an "Access-Request" containing such Attributes as the user's name, the user's password, the ID of the client and the Port ID which the user is accessing. When a password is present, it is hidden using a method based on the RSA Message Digest Algorithm MD5.

The Access-Request is submitted to the RADIUS server via the network. If no response is returned within a length of time, the request is re-sent a number of times.

Once the RADIUS server receives the request, it validates the sending client. A request from a client for which the RADIUS server does not have a shared secret MUST be silently discarded. If the client is valid, the RADIUS server consults a database of users to find the user whose name matches the request. The user entry in the database contains a list of requirements which must be met to allow access for the user. This always includes verification of the password, but can also specify the client(s) or port(s) to which the user is allowed access.

If any condition is not met, the RADIUS server sends an "Access-Reject" response indicating that this user request is invalid. If desired, the server MAY include a text message

in the Access-Reject which MAY be displayed by the client to the user. No other Attributes (except Proxy-State) are permitted in an Access-Reject.

If all conditions are met and the RADIUS server wishes to issue a challenge to which the user must respond, the RADIUS server sends an "Access-Challenge" response. It MAY include a text message to be displayed by the client to the user prompting for a response to the challenge, and MAY include a State attribute.

If the client receives an Access-Challenge and supports challenge/response it MAY display the text message, if any, to the user, and then prompt the user for a response. The client then re-submits its original Access-Request with a new request ID, with the User-Password Attribute replaced by the response (encrypted), and including the State Attribute from the Access-Challenge, if any. Only 0 or 1 instances of the State Attribute SHOULD be present in a request. The server can respond to this new Access-Request with Access-Accept, an Access-Reject, or another Access-Challenge.

If all conditions are met, the lists of configuration values for the user are placed into an "Access-Accept" response. These values include the type of service (for example: SLIP, PPP, Login User) and all necessary values to deliver the desired service. For SLIP and PPP, this may include values such as IP address, subnet mask, MTU, desired compression, and desired packet filter identifiers. For character mode users, this may include values such as desired protocol and host.[4]

4.3 Extensible Authentication Protocol

The Extensible Authentication Protocol (EAP) was originally developed as an extension to PPP allowing for deployment of arbitrary network access authentication mechanisms. With EAP, each PPP peer negotiates to perform EAP during the connection authentication phase. When the connection authentication phase is reached, the peers negotiate the use of a specific EAP authentication scheme known as an EAP type.[5]

Once the EAP type is agreed upon, EAP allows for an open-ended exchange of messages between the access client and the authenticating server (the RADIUS server) that can vary based on the parameters of the connection. The conversation consists of requests for authentication information and the responses. The length and detail of the authentication conversation is dependent upon the EAP type.

In addition to support within PPP, EAP is also supported within the IEEE 802 link layer. IEEE 802.1X, an IEEE standard for network port authentication defines how EAP is used for authentication by IEEE 802 devices, including IEEE 802.11b wireless access points and Ethernet switches. IEEE 802.1X differs from PPP in that only EAP authentication methods are supported. As a result, it is not possible to negotiate the use of PAP with IEEE 802.1X.

As stated above, our approach to WLAN Security provides centralized and mutual authentication. This is achieved by the use of 802.1x/EAP.

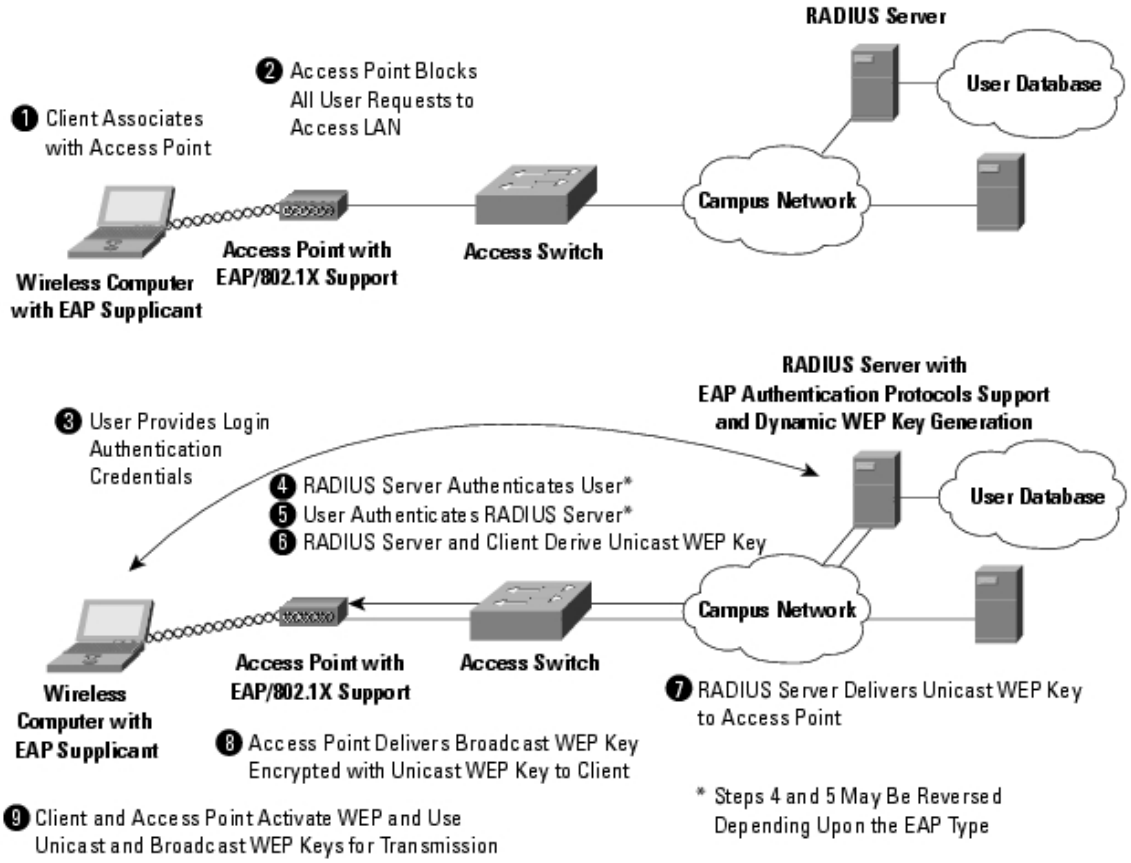
The three main elements of an 802.1X and EAP approach follow:

- Mutual authentication between client and authentication (Remote Access Dial-In User Service [RADIUS]) server

– Encryption keys dynamically derived after authentication
– Centralized policy control, where session time-out triggers
– Reauthentication and new encryption key generation

When these features are implemented, a wireless client that associates with an access point cannot gain access to the network until the user performs a network logon. After association, the client and the network (access point or RADIUS server) exchange EAP messages to perform mutual authentication, with the client verifying the RADIUS server credentials, and vice versa. An EAP supplicant is used on the client machine to obtain the user credentials (user ID and password, user ID and one-time password [OTP], or digital certificate). Upon successful client and server mutual authentication, the RADIUS server and client then derive a client-specific WEP key to be used by the client for the current logon session. User passwords and session keys are never transmitted in the clear, over the wireless link.

The sequence of events follows (refer to Figure):



– A wireless client associates with an access point.
– The access point blocks all attempts by the client to gain access to network resources until the client logs on to the network.
– The user on the client supplies network login credentials (user ID and password, user ID and OTP, or user ID and digital certificate) via an EAP supplicant.

- Using 802.1X and EAP, the wireless client and a RADIUS server on the wired LAN perform a mutual authentication through the access point in two phases. In the first phase of EAP authentication, the RADIUS server verifies the client credentials, or vice versa. In the second phase, mutual authentication is completed by the client verifying the RADIUS server credential, or vice versa.
- When mutual authentication is successfully completed, the RADIUS server and the client determine a WEP key that is distinct to the client. The client loads this key and prepares to use it for the logon session.
- The RADIUS server sends the WEP key, called a session key, over the wired LAN to the access point.
- The access point encrypts its broadcast key with the session key and sends the encrypted key to the client, which uses the session key to decrypt it.
- The client and access point activate WEP and use the session and broadcast WEP keys for all communications during the remainder of the session or until a time-out is reached and new WEP keys are generated.
- Both the session key and broadcast key are changed at regular intervals. The RADIUS server at the end of EAP authentication specifies session key time-out to the access point and the broadcast key rotation time can be configured on the access point.[6]

This approach provides **three significant benefits** over the 802.11 Security:

- The first benefit is the mutual authentication scheme. This scheme effectively eliminates "man-in-the-middle (MITM) attacks" introduced by rogue access points and RADIUS servers.
- The second benefit is a centralized management and distribution of encryption keys. Even if the WEP implementation of RC4 had no flaws, there would still be the administrative difficulty of distributing static keys to all the access points and clients in the network. Each time a wireless device was lost, the network would need to be rekeyed to prevent the lost system from gaining unauthorized access.
- The third benefit is the ability to define centralized policy control, where session time-out triggers reauthentication and new key derivation.[6]

4.3.1 EAP Authentication Protocols

Following are the various authentication protocols used with EAP:

- **LEAP:** Lightweight Extensible Authentication Protocol, or LEAP, is a proprietary implementation by Cisco Systems. With LEAP, mutual authentication relies on a shared secret, the user's logon password, which is known by the client and the network. The RADIUS server sends an authentication challenge to the client. The client uses a one-way hash of the user-supplied password to fashion a response to the challenge and sends that response to the RADIUS server. Using information from its user database, the RADIUS server creates its own response and compares that to the response from the client. When the RADIUS server authenticates the client, the process repeats in reverse, enabling the client to authenticate the RADIUS server. When this is complete, an EAP-Success message is sent to the client and both the client and the RADIUS server derive the dynamic WEP key.

- **PEAP:** PEAP uses a digital certificate for server authentication. For user authentication, PEAP supports various EAP-encapsulated methods within a protected TLS tunnel.
- **EAP-MD5:** EAP-MD5 is a IETF open standard, but offers minimal security. The MD5 cipher is vulnerable to dictionary attacks, and as used in EAP does not support dynamic WEP.
- **EAP-TLS:** EAP-TLS uses digital certificates for both user and server authentication and supports the three key elements of 802.1X/EAP mentioned previously. The RADIUS server sends its certificate to the client in phase 1 of the authentication sequence (server-side TLS). The client validates the RADIUS server certificate by verifying the issuer of the certificate?a certificate authority server entity?and the contents of the digital certificate. When this is complete, the client sends its certificate to the RADIUS server in phase 2 of the authentication sequence (client-side TLS). The RADIUS server validates the client's certificate by verifying the issuer of the certificate (certificate authority server entity) and the contents of the digital certificate. When this is complete, an EAP-Success message is sent to the client and both the client and the RADIUS server derive the dynamic WEP key.
- **EAP-TTLS:** EAP-TTLS provides secure user authentication, using a TLS tunnel to encrypt password-based credentials that would be otherwise subject to dictionary attack on the wireless link. It provides strong security, while supporting legacy password protocols, enabling rapid deployment against your existing security infrastructure.[5]

4.3.2 Benefits of EAP-TTLS

EAP-TTLS provides the following benefits.

- **Completely protects connection credentials from attack:** EAP-TTLS provides complete security for users? connection credentials (i.e., user name and password) as they?re being authenticated to the network. With EAP-TTLS, a WLAN user?s identity and password-based credentials are tunneled during authentication negotiation, and are therefore not observable in the communications channel. This strong security prevents dictionary attacks, man-in-the-middle attacks, and hijacked connections by wireless eavesdroppers ? and protects the network from the havoc an attacker who?s connecting with valid credentials can wreak. EAP-PEAP and EAP-TLS also provide this high level of credential security; LEAP does not. With LEAP, passwords which are short or insufficiently random are vulnerable to dictionary attack.
- **Supports all password protocols, for compatibility with existing authentication scheme:** EAP-TTLS supports all major password protocols, including PAP, CHAP, MS-CHAP, MS-CHAP-V2, EAP-MD5Challenge, and EAP-TokenCard. So, with EAP-TTLS, WLAN users can safely connect using the connection credentials they?re accustomed to using.This siimplifies the access process of WLAN users.Neither EAP-TLS, EAP-PEAP, nor LEAP offers this level of compatibility with existing authentication schemes.
- **Does not require the use of client certificates:** Unlike EAP-TLS, EAP-TTLS does not require the use of client certificates to provide strong credential security. EAP-TTLS and EAP-TLS are similar in that both use TLS (Transport Layer Security, the successor to SSL) as the underlying strong cryptography. However, EAP-TTLS differs in that only the RADIUS servers, not the users, are required to have

certificates. The user is authenticated to the network using ordinary password-based credentials, whose use is made proof against active and passive attack by enclosing it in the TLS security wrapper. Users of EAP-TTLS are, therefore, spared the administrative burden associated with setting up and maintaining a certificate infrastructure.

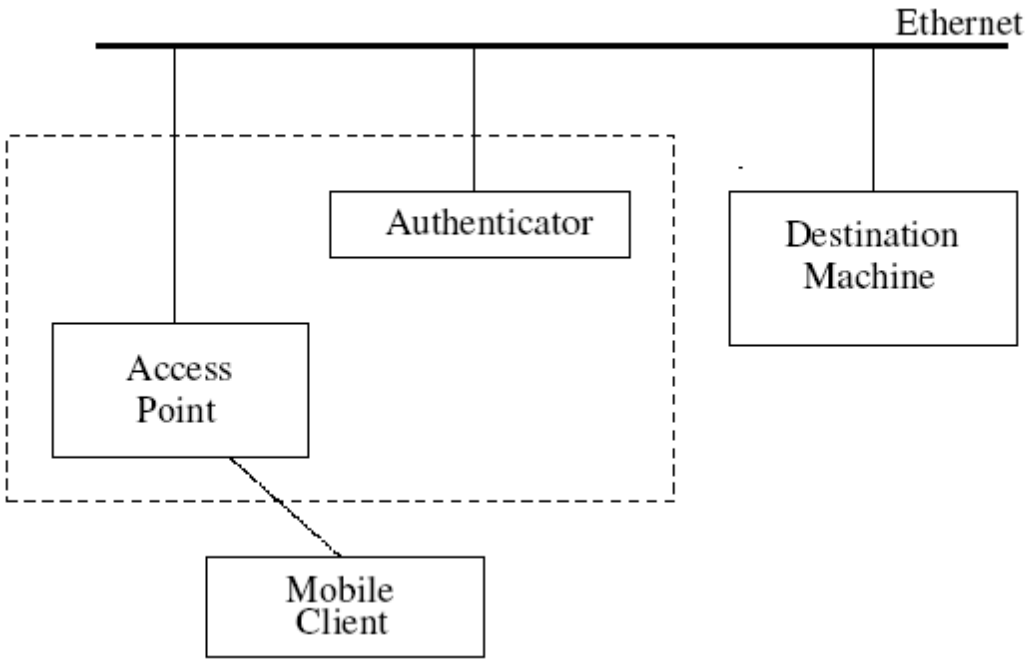
– **Provides data security, plus strong mutual authentication of client and server:** Beyond its strong credential security and ease of management, EAP-TTLS provides additional security techniques to further protect the security of a WLAN user?s connection. With EAP-TTLS, dynamic per-session keys are generated to encrypt the wireless connection and protect data privacy. Frequent re-keying thwarts known attacks against the encryption method used in wireless communications (WEP). In addition, EAP-TTLS provides strong mutual authentication of Client and Server, preventing an intrusion onto the network by an unauthorized user, and ensuring that the client is connecting to the right server.

  With its strong security and compatibility with existing authentication databases and infrastructure, EAP-TTLS puts secure WLAN authentication within any organization's reach.[6]

5 Implementation Details and Testing

Due to their benefits over the other existing protocols as described above, RADIUS Protocol (for Authenticator Server) and IEEE802.1x/EAP-TTLS (as WLAN Protocol) are our protocols of choice for the Authenticating and Security Mechanism.

What follows is the setup of our Mechanism:



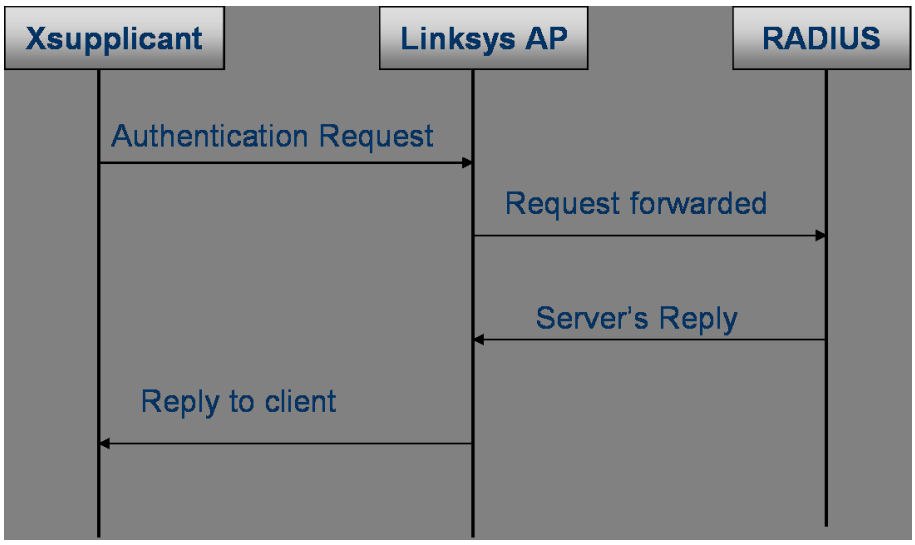
The Authenticator with the Access Point forms the Authenticating System. This is shown by the dotted square in the above figure. The Authenticator, as depicted in the figure, resides on the wired network. It provides the necessary authenticating mechanism to authenticate the mobile client i.e. Xsupplicant wishing to communicate with the Destination Machine.

5.1 Hardware and Software Details

- **FreeRADIUS Server:** FreeRadius v1.0.0 OpenSSL openssl 0.9.7a-23
- **Access Point:** Linksys WirelessG WAP54G Access Point
- **XSupplicant:** xsupplicant v1.0 OpenSSL openssl-0.9.7a-23

5.2 Authentication Process

The Authentication Procedure is depicted in the above figure. The Authentication Request of the XSupplicant is forwarded by the AP to the RADIUS Server which, by referring to its internal database, accepts the user or rejects the user. The corresponding respose of the RADIUS server is sent to user. An authenticated user only can access the Ethernet Resourses.



5.3 Configuration

Files Modified:

- **client.conf:** The secret shared between AP and Authenticator is stored in this file.
- **users:** RADIUS stores user information in this file. new users are added by us and their service parameters are set.
- **huntgroups:** This file stores the information of huntgroups. A huntgroup is formed by a NAS(Network Access Server) and a set of its ports that are being referred by RADIUS Server.
- **radusd.conf:** This is RADIUS Autheticator Configuration file and is used to set various server parameters.
- **eap.conf:** The use of EAP is configured from this file by activating EAP-TTLS Module.

On Access Point, Gateway configuration is performed. The Gateway of AP is set to RADIUS Server.

Installation and configuration of xsuppicant is done on client side.

5.4 Testing and Results

Various EAP variants are tested using RADIUS Server and remote login procedures. A GUI is used on remote machine to display the results of Authentication Procedure.

6 Conclusion

The vulnerabilities of various authentication protocols are studied by us and are documented in this report. Our study reveals that the best approach towards the design of an Authetication and Security Mechanism is a combination of RADIUS Protocol and IEEE 802.1x/EAP-TTLS Protocol. The setup described by us thus provides strong mutual authentication and complete security of ethernet resourses from malicious users. It also mitigates various attacks, including Man in the Middle attack, Session Hijacking and dictionary attacks.

6.1 Future Work

Future work on the setup described here would be actual deployment of the system in CSE Department to impart security to the existing WLAN Setup.

References

[1] Zahur Y. and Yang T.A., *"Wireless LAN Security and Laboratory Designs"*, JCSC 19,3 January 2004

[2] www.cisco.com, *"A Comprehensive Review of 802.11 Wireless LAN Security"*

[3] Mishra A. and Arbaugh W.A., *"An Initial Security Analysis of The IEEE 802.1x Standard"*

[4] RADIUS Protocol rfc *www.freeradius.org/rfc/rfc2865.html*

[5] www.en.wikipedia.org

[6] www.cisco.com *Wireless LAN Security in Depth*

[7] www.cisco.com *802.1x and EAP based Authentication Across Congested WAN Links*

[8] www.microsoft.com *RADIUS Protocol Security and Best Practices*