INTRODUCTION

Recently traditional wired LANs have been replaced by the deployment of wireless LANs (WLAN) as the need for corporate and recreational mobility, flexibility, and increased productivity have increased. This increase is quantified in a report by Cisco systems on WLANs entitled ‘2003 Wireless LAN Benefits Study. This significant rise in wireless access points has given rise to the glaring security issues with the current 802.11 standard. The standard although a good model for wired networks offers no authentication for wireless network users and offers minimal encryption paving the way for intruders to decode transmissions through the air. These glaring security issues from the use of air as a medium of communications has led to the development of the 802.1X standard based on the Extensible Authentication Protocol (EAP). Given in the next section is an introduction to 802.1x which will build the context of the remaining report.

802.1X – AN INTRODUCTION

Designed with EAP conformity, this standard finally delivers an authentication framework. It uses EAP packets encapsulated with Ethernet packets to allow for an authentication session to ensure only rightful users can gain access to a network and its backbone. The format of the EAP packet will be discussed further along. The 802.1X standard outlines the procedure for authentication and it involves the actions of three players; the supplicant, the authenticator, and the authentication server. The supplicant in this process is the user that wants to be authorized to use the given network, the authentication server is where the actual processing is done, and the authenticator is the machine that acts as a relay between the supplicant and authentication server. In terms of WLAN the authenticator is the wireless access point that is being used.

The protocol that is used in the 802.1x standard is EAP encapsulation over LANs (EAPOL). This is the protocol that is used to transmit the EAP packets between the authenticator and the supplicant over the WLAN. Figure 1 is the EAPOL data packet format which outlines how an EAP packet can be encapsulated and carried. Descriptions of the remaining fields of the EAPOL are omitted as a discussion of these fields could comprise a full length report itself, rather the reader for the scope of this report need only focus on the encapsulation of the EAP packet.

<table>
<thead>
<tr>
<th>Ethernet Type</th>
<th>Protocol Version</th>
<th>Packet Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Packet Body</td>
</tr>
<tr>
<td>Packet Body Length</td>
<td></td>
<td>Packet Body</td>
</tr>
</tbody>
</table>

The grayed field Packet Type is where it is determined if an EAP packet is being carried. If the value is 0000 0000 then the frame carries an EAP packet which is contained in the Packet Body. It is from the Packet Body field where the supplicant/authenticator will
strip the EAP packet and examine the contents. Other values in the Packet Type can be used to signal various types of EAPOL frames but we will not concern ourselves with these.

**EAP – Packet By Packet**

The figure below outlines the basic datagram for EAP use with a description of each field in the datagram following Figure 2.

<table>
<thead>
<tr>
<th>CODE</th>
<th>IDENTIFIER</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2 - EAP Data Packet**

The code field can take one of four different values depending on the source of the data packet. The codes are 1, 2, 3, and 4, for Request, Response, Success, and Failure respectively. The identifier field is used to make sure that the correct responses from the authentication server or supplicant are for the appropriate authentication session by matching the identifier fields. The length field is similar to other length fields seen throughout the course and lets the recipient know the length of EAP packet. Finally the data field contains the contents of the message and depends on the code field for the type of content to be sent.

**Request and Response Packets:**

<table>
<thead>
<tr>
<th>Code</th>
<th>Identifier</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type-Data</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3 - Request and Response Datagram**

The request and response packets in Figure 3 have two more specific fields within the framework of the general EAP packet, which are Type and Type-Data. Here the Type field can take the value of 1...6 for the types Identity, Notification, Nak, MD5-Challenge, One-Time Password, and Generic Token Card respectively. Given next is a brief description of the different types mentioned above.

The Identity type is usually used by the authenticator to find out the identity of the user trying to gain access. The response of the user must contain the same type as the request, and the actual identity must be returned in the Type-Data field or left empty if the identity is not known. The Notification type is used to display imperative messages to the user such as time warnings or other concerns from the authenticator. The Nak type is only used by the user to respond to the authenticator. It is used when the user does not agree on the authentication type and requests another type be used by specifying the type in the Type-Data field.

The remaining three types given above are all authentication methods that can be used if agreed upon by the user and authenticator. MD5-Challenge is a handshake authentication protocol, One-Time Password uses a challenge system where the user is
challenged to enter one of the correct dictionary words, while Generic Token Card requires the user to return the correct token card information for authentication.

Success and Failure Packets:

<table>
<thead>
<tr>
<th>Code</th>
<th>Identifier</th>
<th>Length</th>
</tr>
</thead>
</table>

Figure 4 - Success and Failure Datagram

The success and failure packets given in Figure 4 only transmit the first three fields. Depending on whether the Code is 3 for success or 4 for failure the user will be notified. No response is required by the user to the given packet.

**AUTHENTICATION PROCEDURE**

Association is first required before the authentication procedure can start. Once association has been accomplished the authentication procedure follows the skeleton format given subsequently. The authenticator will first send the initial request packets to the user. These request packets will be sent continuously until a response packet with the same Identifier is received or the number of retries expires. Next, once a request packet has been received the user will send a response packet to the authenticator with the required information and the correct identifier. Response packets are only sent when requests have been received and only one response is issued for multiple requests of the same identifier. This is because the authenticator might have issued multiple requests due to latency or its timer values and only requires one response, while the others should be disregarded. The authenticator will then send this packet to the authentication server who will respond with a challenge of one of the types given earlier; MD-5, OTP, Token Card.

Next the authenticator will encapsulate the challenge into an EAPOL packet and sends it to the supplicant. Finally once the response has been received by the authenticator, it will send the response on to the authentication server. The authentication server will decide whether access will be allowed by issuing a success or failure packet depending on the information received. One has to remember that this framework is just outlining the basic steps required for authentication, while in application contexts the request and response phase might be more complex as requests with specified protocols might be denied by the user using the Nak type for a more desired authentication protocol. So one can see how this process can become complicated as easily as it can be made basic.

**CONCLUDING REMARKS**

The introduction of the 802.1x standard and its use of the EAPOL protocol have effectively helped ensure secure authentication of users to networks through wireless access points. It is the desire of this report to motivate the reader to improve/design WLAN security using a similar approach with a basic understanding of 802.1x, EAPOL, and EAP given in this report.
ENDNOTES


