International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

Analysis on the open security issues in 802.1x EAP security standard (RFC3748)

Jani Ahamed Habeeb

Software Engineer, India

Abstract – *In today's internet world, there is a lot of data traffic that gets exchanged over the network. How secure is your system from malicious packets or security threats is the biggest challenge that is in front of us. In order to keep the system safe and secure from security threats, most organizations implement the 802.1x EAP (dot1x – RFC 3748) security standard. There are in fact a lot of security loopholes even if we implement this 802.1x security standard. We will analyze the security issues that are open and unaddressed in 802.1x which might lead to getting our systems exposed to malicious threats and also get to know the best practices to address these security loopholes.*

Key Words: 802.1x, RFC 3748, security loopholes, threat prevention, EAP, need for security

1. INTRODUCTION

Extensible Authentication Protocol (EAP), an authentication framework that runs directly over data link layers such as Point-to-Point Protocol (PPP) or IEEE 802 without requiring IP. EAP may be used on dedicated links, as well as switched circuits, and wired as well as wireless links. EAP provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees. This security standard is implemented by most organizations to prevent themselves from security threats.

1.1 EAP terminologies

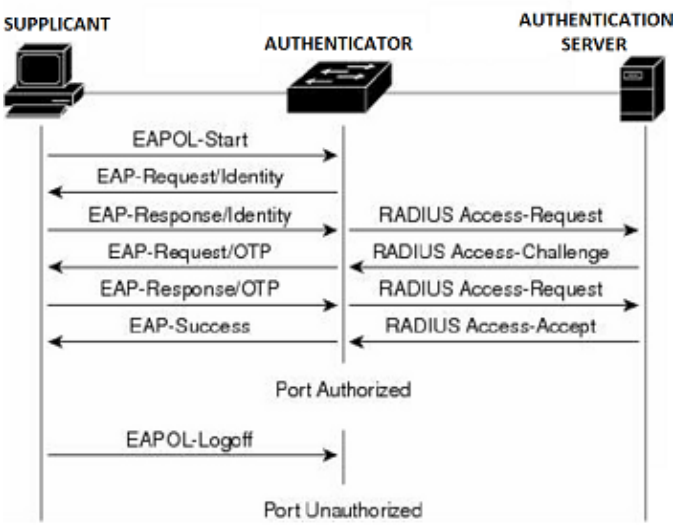
Authenticator: The entity that is on end of the link that initiates the EAP authentication

Supplicant: The entity that is on the end of the link that responds to the authenticator

Authentication server: An entity that is responsible for providing an authentication service to an authenticator.

1.2 EAP Working model

- The supplicant may send an EAP start message.
- The authenticator sends an EAP request identity message.
- The supplicants EAP response packet with the client's identity is proxied to the authentication server by the authenticator.
- The authentication server challenges the client to prove themselves and may send its credentials to prove itself to the supplicant(mutual authentication)
- The supplicant checks the authentication server's credentials and then sends its credentials to the server to prove its identity.
- The authentication server accepts or rejects the clients request for connection.
- If the supplicant was accepted, the authenticator changes the virtual port connected to the supplicant from unauthorized state to Authorized state allowing full access to the network.
- When the supplicant logs off, the supplicant's virtual port is again moved to unauthorized state.



802.1x Authentication Flow

2. Security Issues in the above Model

Though the above model validates the supplicant well before declaring it to be authorized and granting access to network, there is still a possibility of getting exposed to security vulnerabilities.

In the above design, once the authentication server validates the credentials of the supplicant and finds it to be authorized, it conveys it to the authenticator. Now as soon as the authenticator receives the Authorization passed for the Supplicant message from the Authentication server, it moves

the Supplicant connected virtual port from Unauthorized to Authorized State. This grants complete access to the network for the supplicant. Following this, the authenticator sends an EAP SUCCESS message to the Supplicant which will inform the Supplicant that it has been granted access.

The Supplicant has no way to know that he has been authorized and granted access to the network, if the EAP Success message is not received from the Authenticator.

There are two conditions to be analyzed here.

1) What will happen if the EAP Success message is lost in the network and it never reaches the supplicant.

2) What will happen if the supplicant is not alive or has left the network ungracefully without the knowledge of the authenticator.

Since the EAP method completely relies on the lower layer indications, any of the above condition is dangerous.

In scenario 1) The supplicant will never know of its authorization since the EAP success did not reach. The downfall here is that there is no retransmission mechanism involved for EAP success messages and there is no acknowledgement involved from the supplicant to the authenticator as well. So, if an EAP success is lost, it's lost forever.

In scenario 2) the supplicant has left the network ungracefully. The authenticator now has no knowledge about the supplicant's presence. The authenticator might move the virtual port that was connected to the supplicant to Authorized state allowing traffic. Since there is no supplicant connected at the other end, why is this system resource (virtual port) wasted.

There are a lot of security vulnerabilities here.

1) Since the port is now in Authorized state (it is open for all) Attackers or hackers can easily misuse the unattended open traffic port and start launching network attacks.

2) An attacker might initiate as many connections and exhaust all system resources leading to denial of service attacks.

3) Some other hacker system may start sending traffic by spoofing the same MAC as the supplicant who got authenticated

4) If there is an unattended port open to public in any system it becomes vulnerable to all kind of hacker attacks.

5) System resources are limited, so if any system resource like a port is left open for no reason without any proper attention, it might lead to system inefficiency in handling proper supplicants.

3. Possible solutions

Inorder to address the issue of system resource optimization where in a port is not opened without even knowing whether a valid supplicant is connected or not, introduce an Acknowledgement mechanism.

As part of this acknowledgement mechanism, once an authenticator knows that the supplicant is authorized, it sends out an EAP success message to the supplicant and waits for the Acknowledgment. Upon confirmation of the acknowledgement the port can be moved to Authorized state

To solve the issue of EAP success message being lost in the network before reaching the suppliant, we can introduce a retransmission mechanism, where in the authenticator waits for a prescribed time after sending the EAP Success packet for an Acknowledgement. If the Acknowledgement is not received, the authenticator decides that the EAP success did not reach the supplicant and retransmits the same.

To have a more reliable security solution from the EAP perspective, and solve the issue of another hacker compromising the network and making use of unattended Authorized open ports, in the name of valid supplicant by spoofing, we can implement key exchange mechanism as in PEAP over TLS and encrypt the complete traffic flow between the authenticator and the supplicant. With this no other hacker or attacker can spoof or communicate to the authenticator without knowing the master keys used for communication.

4. CONCLUSIONS

Security is of prime importance in any data network and if there is even a slightest chance of vulnerability it is going to question the existence of the security system itself. Day by Day hackers are finding new ways to exploit the network and hence a robust solution is a must to prevent hackers from getting access to the network.

REFERENCES

[1] https://tools.ietf.org/html/rfc3748

BIOGRAPHY

Jani Ahamed Habeeb , a software professional with over 6 years of experience in network security