Penetration Testing of IEEE 802.1X Port-based Authentication Protocols using Kali Linux Hacking Tools

Michael Kyei Kissi
Department of Computer Science
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

Michael Asante, PhD
Department of Computer Science
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

ABSTRACT
Extensible Authentication Protocol (EAP) was designed to provide a general structure for several different authentication methods. IEEE 802.1X uses EAP as an authentication tool. The IEEE 802.1X standard defines a client-server authentication and access control protocol that restricts unauthorized users from connecting to a network. This paper aims at using penetration testing to conduct security assessment of some IEEE 802.1x Port-Based Authentication protocols (PEAP, EAP-TTLS and Inner Authentication Method MSCHAPv2 and PAP). Vulnerabilities identified were exploited using Kali Linux with its Aircrack-ng tools.

Keywords
IEEE 802.1x, EAP, PEAP, TTLS, PAP, MSCHAPv2, Penetration Testing, Wireless Network, WLAN, Kali Linux

1. INTRODUCTION
Network infrastructure security and risk factors are major concerns to safeguard network against attacks. Every network infrastructure is capable of being under attack by intruders [1]. It is of essence for network administrators to have a complete account of manageability, scalability and security of the network to maintain the confidentiality and integrity of organizational data [2]. According to [3], newer vulnerabilities evolves daily due to technological advancement. The IEEE 802.1x protocol also known as Port-based Network Access Control (PNAC) that uses Extensible Authentication Protocol (EAP) was created by IEEE to secure user authentication or verify user's identity when connecting to WLAN [4]. Its standard defines encapsulation methodologies for the transport of EAP over LAN (EAPOL) and gives a powerful authentication framework for high level security [5]. This paper presents a security assessment of some IEEE 802.1x Port-Based Authentication Protocols using penetration testing tools to examine and exploit identified vulnerabilities. Extensible Authentication Protocol (EAP) methods assessed comprise of PEAP, EAP-TTLS and Inner Authentication Method MSCHAPv2 and PAP.

2. LITERATURE REVIEW
The 802.1x standard indicates the operational components, protocols and architecture that assist port-based authentication of users on a network. 802.1x standard objective is to control user access and protect unauthorized transmission [6]. The standard is based on the Extensible Authentication Protocol (EAP) where clients are permitted to dynamically opt for an authentication mechanism. The selected mechanism is based on information transmitted through a Remote Authentication Dial-In User Service (RADIUS) message [7]. WLAN environment that is secured with EAP is known as WPA/WPA2 Enterprise network [8]. The WPA/WPA2 Enterprise networks provides each user an account details (username and password) to access the EAP protocol rather than the use of a passphrase. [8] noted that the ports in port-based authentication are of two states (authorized and unauthorized) which is dependent on the identity of the user. The ports are classified as uncontrolled and controlled port. Access to the network is granted, when the controlled port is open based on a successful authentication of the supplicant as shown in figure 1. In WLAN networks, virtual ports are used since physical ports do not exist.

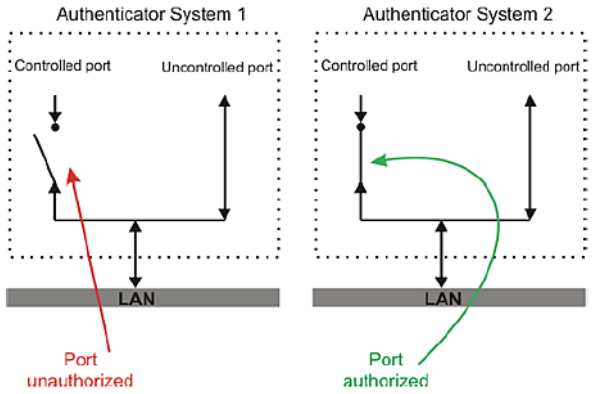

Figure 1: 802.1x Port-Based Authorization [9]

2.1 Communicating Entities
IEEE 802.1x Port-Based Authentication is based on three communicating Port Access Entities (PAEs). These are the Supplicant, Authenticator and Authentication Server (AS) [8].

- Supplicant known as client that connects to the Authenticator and are being authorized. Data packets are only transmitted by the supplicant to the network after it has been authenticated to the port. Supplicants includes devices such as smartphones, laptops, and other Wi-Fi devices.
- Authenticator also known as Network Access Server (NAS) operates as a negotiator between the Supplicant and Authentication server during authentication. When authentication is successful, a secure transmission channel is created to enable access of network resources to the supplicant.
- Authentication Server (AS) does the main authentication of the supplicant using the RADIUS server.

Figure 2 shows a successful authentication of the Supplicant having access to network resources. It also shows the protocols involved in the 802.1x authentication.

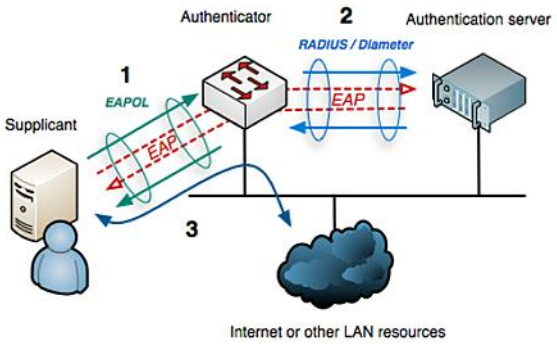

Figure 2: 802.1x Network Communication Entities and Protocols [10]

2.2 Extensible Authentication Protocol (EAP)

The protocol gives a foundation for network supplicants and authentication servers. It is the most used authentication method as compared to other wireless security protocols [11]. The EAP framework supports diverse kinds of port authentication methods known as EAP methods. It operates above the data link layer therefore no IP is required in order to function. Initially the EAP was used exclusively over PPP or Ethernet wired network but was later extended to the 802.11 wireless networks [11] [12]. The supplicant and Authenticator exchange EAP messages known as LAN messages and also RADIUS messages exchanged between the AS and Authenticator.

2.2.1 EAP Over LAN Message Types

There are four different types of EAP messages that occur between the Supplicant and Authenticator [11]. These are:

- EAP Request Message: A request message that the supplicant uses to prove its identity sent by the authenticator.
- EAP Response Message: The authenticator receives the response message sent by the supplicant to prove its identity.
- EAP Success Message: A success message is sent by authenticator to the supplicant when the AS accepts the identity of the supplicant.
- EAP Failure Message: A failure message is sent by authenticator to the supplicant when the AS rejects the identity of the supplicant.

2.3 Remote Authentication Dial In User Service (RADIUS)

RADIUS is installed and configured in the AS of most enterprise networks [8]. The RADIUS is responsible for saving information about user credentials on the server to authenticate the supplicant when it wants to connect to the network [13]. The Authenticator and the RADIUS exchange attribute refer to as Attribute Value Pairs (AVPs) for Accounting, Authentication and Authorization (AAA) [14] [15].

2.3.1 EAP Over RADIUS Message Types

There are four different kinds of EAP messages that occur between the Authenticator and AS [8]. These are:

- RADIUS Access Request Message: An EAP message forwarded by the Authenticator from the supplicant to the AS.
- RADIUS Access Challenge Message: A message relayed by the AS to prove the identity of the supplicant or Authenticator.
- RADIUS Access Accept Message: The AS sends this type of message to the Authenticator when the identity of the supplicant or authenticator is valid or accepted.
- RADIUS Access Reject Message: The AS sends this type of message to the Authenticator when the identity of the supplicant or authenticator is invalid or rejected.

2.4 EAP Authentication Process

As stated by [8], most EAP methods use the same primary procedures to authenticate a Supplicant to the Authentication Server.

Firstly, the Supplicant waits for the Authenticator to transmit an Identity Request packet or EAPOL Start Packet. The supplicant after receiving the Identity Request replies with an Identity Response packet which contains the Network Access Identifier (NAI).

In the next step, The AS sends a Request message to authenticate the supplicant. The Supplicant either accepts or decline by transmitting an EAP-Negative-Acknowledgement (NAK) Response. Authentication process continues when both uses the same EAP method.

The AS generates a Master Session Key (MSK) outlined by the EAP method once the port authentication is acknowledged. The AS computes the PMK from the MSK and sent over a secure channel to the Authenticator. Also, the Supplicant derives its PMK from the MSK. A four-way handshake occurs between the supplicant and Authenticator to compute the PTKs after the supplicant receives an EAP-Success packet from the Authenticator. When there is an authentication failure, EAP-Failure packet is sent and the port remains unauthorized. The EAP authentication procedure is summarized in figure 3.

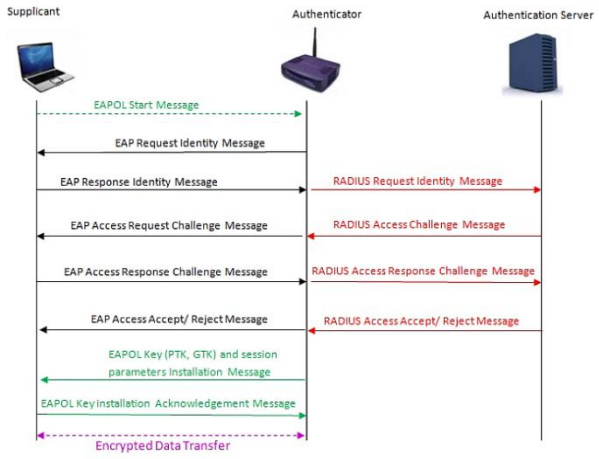

Figure 3: EAP Authentication Procedure [17]

2.5 EAP Authentication Methods

EAP authentication are based on EAP methods depending on the Type field of the EAP Requests and Reponses. The authentication happens between the Supplicant and Authentication Server once EAP Identity occurs [8] [16]. EAP

methods that are popular and mostly used include, LEAP, EAP-TLS, EAP-TTLS and PEAP [17]. Some of these EAP methods are briefly explained.

- EAP-Generic Token Card (GTC): EAP-GTC is one of the simplest EAP methods where the Authenticator sends Request message and the supplicant replies with a Response message. Message exchange between the two entities are sent in plaintext read from a token card. Since all packets (EAP Request and Response) are sent in plaintext, an attacker can eavesdrop the credentials of users [11] [8].
- Lightweight Extensible Authentication Protocol (LEAP): LEAP, an EAP authentication which uses the Cisco proprietary protocol. LEAP supports WEP keys generated dynamically and key rotation to improve network security [18]. In spite of the security mechanisms of LEAP, data packets are transmitted in plaintext which makes it easier for an attacker to sniff and capture.
- EAP-Transport Layer Security (TLS): Authentication in EAP-TLS is based on a TLS handshake. A mutual authentication occurs between the supplicant and AS by an exchange of digital certificate signed by a Certificate Authority (CA). The supplicant sends a certificate to authenticate AS and the AS also sends a certificate authenticate the supplicant. In wireless network, TLS can be used as an authentication protocol using digital certificates and also for transmission of legacy authentication protocols through a secure tunnel [19] [8].
- Protected EAP (PEAP): PEAP establish a TLS tunnel between the Supplicant and AS. PEAP authentication is based on TLS where the AS only authenticate the supplicant and not vice versa. The AS requires a public key certificate (Server-side certificate). After the supplicant authenticate the identity of the AS, series of EAP messages are exchanged which is encapsulated inside TLS messages. TLS session keys derived by the supplicant and AS are used to encrypt and authenticate TLS messages. Since the method only uses Server-side certificate, an attacker can issue a fake certificate, if the supplicant accepts, the AS is hacked [20] [8].
- EAP-Tunneled Transport Layer Security (TTLS): Simon Blake-Wilson and Paul Funk originated the EAP-TTLS method which is defined in the RFC 5281 [8]. The EAP method permits the use of legacy authentication protocols (such as CHAP, MSCHAPv2, MS-CHAP and PAP) inside a TLS tunnel. EAP-TTLS and PEAP are similar except that, PEAP only allows MSCHAPv2 as inner authentication method. Also, EAP-TTLS does not support cryptographic binding or method chaining. A TLS channel is created in EAP-TTLS to exchange AVPs to validate user credentials against any legacy authentication protocols whilst in PEAP, the TLS channel is used to protect a second EAP exchange known as the "inner" EAP exchange [19] [8] [21].

2.5.1 EAP Inner Authentication Methods

EAP as the foundation of wireless network security is based on TLS that has same objective as the Secure Socket Layer (SSL) protocol to establish secure encryption and authentication channel (tunnel) over untrusted networks. The tunneled method is mostly called the "inner" authentication method. Inner authentication methods are tunneled through TLS because they prevent the supplicant from knowing the identity of the AS. This makes them susceptible to MITM attack [22]. Some of these inner authentication methods are briefly explained.

- Password Authentication Protocol (PAP): PAP defined in RFC 1334 was originally designed to operate with PPP. User information (username and password) are traversed in plaintext over the network. PAP is recommended for networks which has its own privacy protection. EAP-TTLS is the only EAP method that supports PAP [22].
- Challenge Handshake Authentication Protocol (CHAP): CHAP defined in RFC 1994 was also created to operate with PPP. CHAP is based on a challenge handshake where the authentication server challenges the supplicant to prove its possession of the shared secret by sending a challenge responds. CHAP is only supported by EAP-TTLS [22].
- Microsoft CHAP (MS-CHAP): Microsoft implemented MS-CHAP with advanced effectiveness and efficiency in Windows systems as compared to CHAP where shared secret information is saved encrypted at both stations. Passwords stored on the server are hashed with a specific one-way cryptographic hash. The hash method used by the server is known to the client where it produces a "Matching" password which is used in a challenge-response handshake authentication. The client is authenticated by attesting to the hashed value of the password. MS-CHAP is only supported by EAP-TTLS [22].
- Microsoft CHAP version 2 (MSCHAPv2): MSCHAPv2 was also designed by Microsoft to address the shortcoming of MS-CHAP where password encryption was weak. MSCHAPv2 provide mutual authentication and improve keying and key generation [22]. The supplicant asks for an Authenticator Challenge (CS) from the AS which is randomly generated of size 16-byte. The Supplicant response to the challenge by generating Peer Authenticator Challenge (CC) of size 16-byte. The Username, CC and CS from the AS are all concatenated and hashed by the supplicant to generate an 8-byte Challenge (C). The supplicant uses the Windows NT hash similar to MD4 hash to generate NT password hash (NTHash) by hashing its password and is padded with 5 bytes of zeros to generate three Data Encryption Standard (DES) each of 7-byte size. A parity bit is attached to each 7 bytes to form an 8-byte long DES key (K1, K2, K3). Each key is used to encrypt the Challenge Hash generated by the supplicant. The three Challenge Hashes are concatenated to form a 24-byte Challenge Response (R). The Challenge Response, Username and CC are transmitted to the AS. The AS decrypts the received response the NThash of the supplicant save in a database. When the decrypted response corresponds to the challenge, a positive authenticator response is sent to the supplicant. The AS uses the NThash and the CC from the supplicant to generate a 20-byte Authenticator Response. The supplicant then generates its Authenticator Response and compares with the Response from the AS, if they match the Supplicant authenticates the AS [23] [24] [25] [26] [27]. MSCHAPv2 is supported by TTLS and PEAP [22]. Figure 4 shows the generation of the Challenge Response process. The authentication process can be deduced as:

  C = SHA1(CS, CC, Username)
  NTHash = MD4(Password)
  K1 | K2 | K3 = NTHash | 5 byte of 0
  R = DES(K1,C) | DES(K2,C) | DES(K3,C)

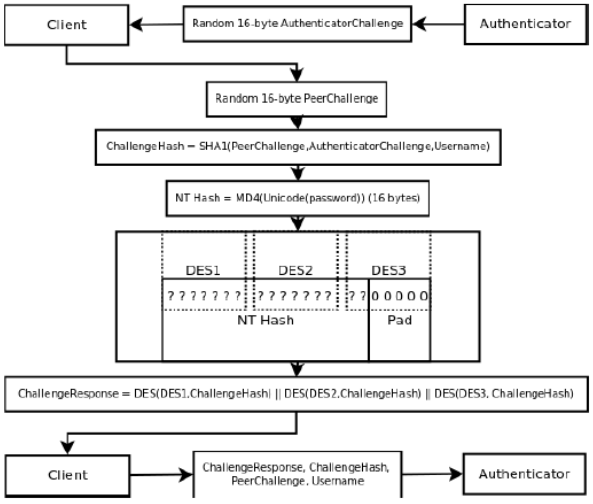

Figure 4: Generation of the Challenge Response [28]

2.6 Attacks on EAP

According to [8], flaws exist in the design of some EAP methods and its usage in wireless environment would be a major security risk. These kind of vulnerabilities in the design makes it more attractive to an attacker.

2.6.1 MSCHAPv2 Hash Reuse and Dictionary Attack

The MSCHAPv2 protocol has few weaknesses that makes it vulnerable to attacks. [26] noted that there is weakness in how the NTHash is calculated. The hash is not salted as a result the NTHash is same as the password which makes it easier for an attacker to reuse the hash. The attacker can authenticate with the NTHash as the user and impersonate the AS. Rainbow tables can be used by the attacker to crack the password since the NTHash is not salted.

MSCHAPv2 is based on a challenge-response mechanism sent in plaintext. This makes MSCHAPv2 vulnerable to dictionary attack. An attacker can manage to sniff an MSCHAPv2 exchanged messages or handshake by executing MITM attack on the user. The 16-byte Random challenge from the AS, Peer Authenticator Challenge, Challenge Response and Username are known to the attacker. In computing the Challenge Response, the only unknown element to the attacker is the NT password hash (NTHash). The attacker can perform a dictionary attack by iterating guessed passwords using a wordlist or dictionary to generate the NTHash value and recalculate the Challenge Response. If the sniffed challenge response matches the attacker's calculated challenge response, means the guessed password is correct [25] [29] [8].

3. METHODOLOGY

The penetration test was conducted by using an experimental WLAN laboratory setup. The study considered to use the network laboratory in order not compromise any individual or organization network due to privacy and legality of user information.

3.1 Laboratory Experiment Setup and Requirements

The experiment required the use of an Authenticator (wireless router), Authentication Server, an external wireless adapter and two laptops (one as the PenTester PC and other as the Supplicant, the supplicant could be any device with wireless connectivity). The IEEE 802.11 encryption protocols were configured on the Authenticator. FreeRadius-WPE was installed and configured as the Authentication Server in Kali Linux OS as a virtual host machine to assess the 802.1x port-based authentication. An external wireless card (Alfa AWUS036H) that supports packet sniffing and injection was used to connect to the PenTester PC to be used with the Kali Linux since the internal wireless card of the PC was inaccessible on the virtual host and does not support packet sniffing and injection. Figure 5 illustrate the connections of the used devices.

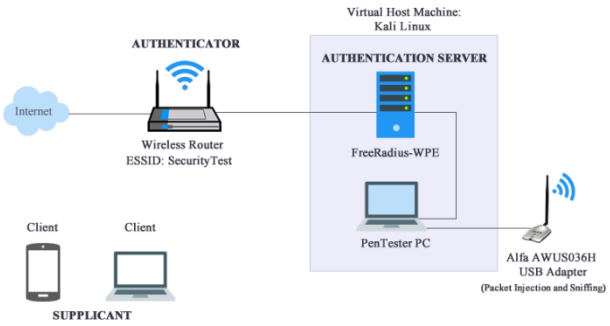

Figure 5: Diagram of Penetration Test Experiment Setup

3.2 Exploiting Vulnerabilities in PEAP and EAP-TTLS Using Inner Authentication Method MSCHAPv2

PEAP and TTLS both uses TLS to protect legacy authentication protocols from interception and requires a certificate on the RADIUS server for the supplicant to validate server identity. Vulnerabilities were discovered and exploited in the authentication protocol include the following:

3.2.1 Supports only Server-Side Certificates

PEAP and EAP-TTLS uses only server-side certificates (Public Key Certificate) to confirm the credentials of the supplicant. The identity of the server is unknown and cannot be validated by the supplicant. Rogue server setup by the attacker was used to issue fake server-side certificates, the supplicant blindly accepts the fake certificate, the AS is hacked.

3.2.2 NTHash (MSCHAPv2 Hash value) can be computed by the Attacker

MSCHAPv2 is uses a challenge-response mechanism. In computing the Challenge Response, the only unknown element to the attacker is the NT password hash (NTHash). The 16-byte Random challenge from the AS (CS), Peer Authenticator Challenge (CC), Challenge Response (R) and Username are known to the attacker. An attacker can iterate or brute force guessed passwords using a wordlist or dictionary to generate the NTHash value and calculate the Challenge Response. The attacker then compares its computed challenge response with the one originally captured. The Challenge Response is computed as:

C = SHA1(CS, CC, Username)
NTHash = MD4(Password)
K1 | K2 | K3 = NTHash | 5 byte of 0
R = DES(K1,C) | DES(K2,C) | DES(K3,C)

With the discovered vulnerabilities as a result of fake digital

certificate and cracking of the password, a penetration test was conducted as a proof of the existence of these vulnerabilities.

3.2.3 Cracking PEAP and EAP-TTLS Using Inner Authentication Method MSCHAPv2

FreeRadius-WPE Server was configured to use PEAP as the default EAP Type to accept MSCHAP as inner authentication method by modifying the "*eap.conf*" configuration file as shown in figure 6. The same configuration settings were used for the EAP-TTLS since they all use the MSCHAP inner authentication method.

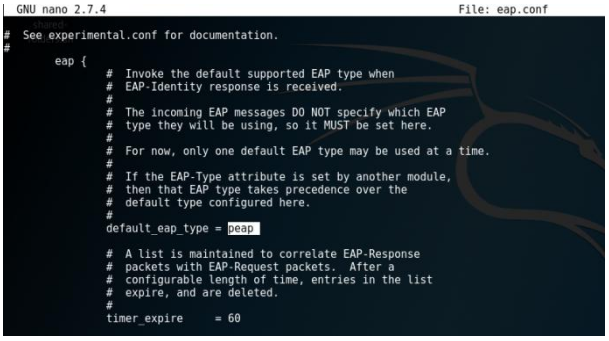

Figure 6: Configuration of PEAP and EAP-TTLS to accept MSCHAPv2 Inner Authentication Method

The FreeRadius-WPE Server was configured to issue fake digital certificates as shown in Figure 7 using the make certificate command "*./bootstrap*".

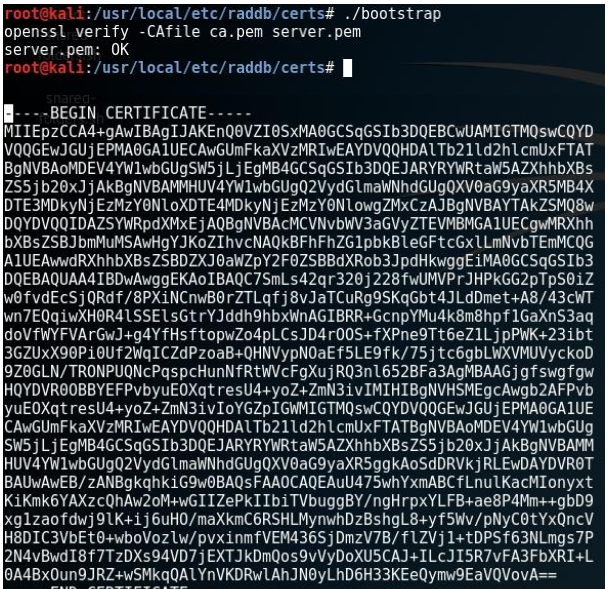

Figure 7: Generation of Fake Server-Side Digital Certificate

The FreeRadius-WPE Server was started to authenticate and validate supplicant credentials before giving access to the network. The command "*radius –s –X*" as shown in figure 8 was used.

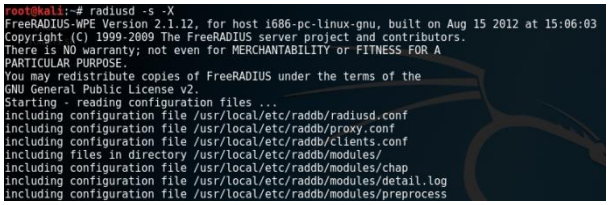

Figure 8: Starting FreeRadius-WPE Server

The command "*tail -f*" was used to log all the MSCHAP handshake interactions between the supplicant and the AS by using the "*freeradius-server-wpe.log*" file as shown in figure 9.

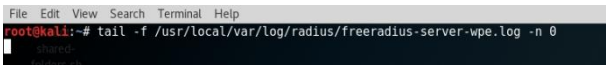

Figure 9: Capture of MSCHAP Handshake using Freeradius-server-wpe.log file

A legitimate client was connected to the Rogue AP by providing the required credentials (username and password) as shown in figure 10 and the fake digital certificate was issued by the server to authenticate the supplicant. The supplicant upon accepting the certificate was attacked.

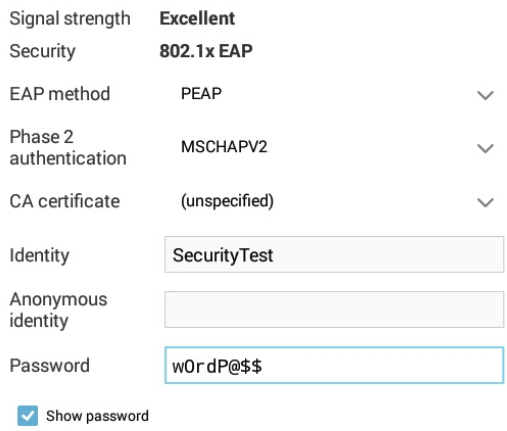

Figure 10: Supplicant Connection Using MSCHAPv2 Inner Authentication Method

The freeradius-server-wpe.log file captures MSCHAP Handshake as soon as the supplicant provides credentials and accepts the fake certificate. Figure 11 shows the captured Username, MSCHAP challenge and Response.

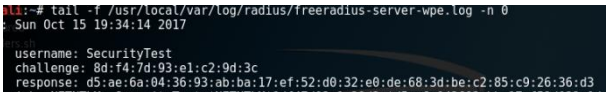

Figure 11: Captured MSCHAPv2 Handshake

Since the Username, MSCHAP challenge and Response has been captured, any attacker could crack the password using dictionary attack. A script called "*Asleap*" developed by Joshua Wright was used to crack the password by providing the captured Challenge and Response with a dictionary file.

The *Asleap* script calculates the MSCHAPv2 hash value for each password in the dictionary file and compares it with the captured MSCHAPv2 response, if there is a match, the

password is cracked. The supplicant's password was successfully cracked since the password (w0rdP@$$) existed in the dictionary file as highlighted in figure 12.

Figure 12: Crack of MSCHAPv2 Password using Asleap

3.3 Vulnerabilities in EAP-TTLS Using Inner Authentication Method PAP

Password Authentication Protocol (PAP) transmits supplicant credentials (username and password) across the network in plaintext (unencrypted). The experiment conducted proves the vulnerability in this authentication protocol.

3.3.1 Cracking EAP-TTLS Using Inner Authentication Method PAP

The FreeRadius-WPE Server was started to authenticate and validate supplicant credentials before giving entry to the network as shown in figure 8. The communication between the AS and the supplicant was eavesdropped using the "freeradius-server-wpe.log" file as shown in figure 9. A connection was made by the supplicant by providing user identity to the wireless network using PAP as the inner authentication method and TTLS as EAP method shown in figure 13.

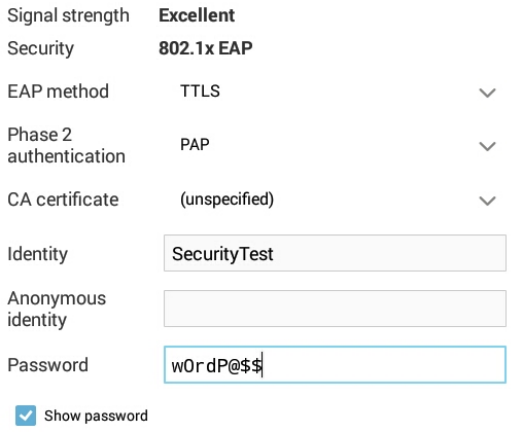

Figure 13: Supplicant Connection Using PAP Inner Authentication Method

Figure 14 shows the username (SecurityTest) and password (w0rdP@$$) of the supplicant that were captured in plaintext without any form of encryption from the freeradius-server-wpe.log file; hence, the PAP inner authentication method was successfully cracked.

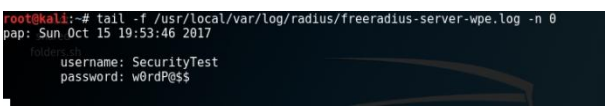

Figure 14: Successful Crack of PAP Inner Authentication Method

4. RESULT ANALYSIS

EAP methods and their inner authentication methods used for this research study were assessed by discovering possible vulnerabilities. The vulnerabilities identified were successfully exploited by the attacker as proof of their existence in the respective authentication protocol.

4.1 Analysis on Vulnerabilities in PEAP and EAP-TTLS Using Inner Authentication Method MSCHAPv2

4.1.1 Supports only Server-Side Certificates

PEAP and EAP-TTLS uses only server-side certificates where the supplicant cannot validate. The supplicant carelessly accepts the fake digital certificate that was created by the attacker as shown in figure 15.

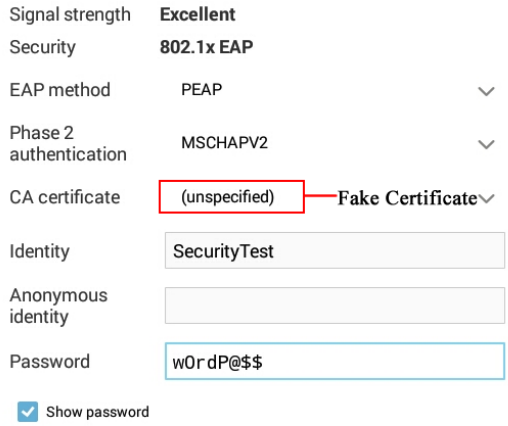

Figure 15: Supplicant Accepts Fake Server-side Certificate

4.1.2 NTHash (MSCHAPv2 Hash value) can be computed by the Attacker

The TLS tunnel was successfully created for an MSCHAPv2 Handshake exchange to occur between the victim's machine and the attacker. The attacker successfully captures the MSCHAPv2 Handshake which contains the challenge response in plaintext as shown in figure 16.

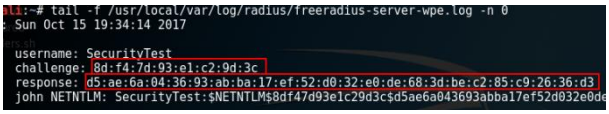

Figure 16: Successful Capture of MSCHAPv2 Challenge and Response

4.1.3 Cracking PEAP and EAP-TTLS Using Inner Authentication Method MSCHAPv2

In computing the Challenge Response, the only unknown element to the attacker is the NT password hash (NTHash). The attacker brute force the password using a wordlist or dictionary to generate the NTHash value and computes the Challenge Response using the Asleap script. The attacker successfully obtains the NTHash value and unencrypted password as highlighted in figure 17.

The outcome of the test experiment indicates that MSCHAPv2 is vulnerable to dictionary attack. The passphrase could be complex since passwords are case

sensitive and special character could be included. An attacker will be able to crack MSCHAPv2 depending on the dictionary, if the password exists in the wordlist, then it will be successfully cracked.

Figure 17: Successful Crack of PEAP and EAP-TTLS Passphrase using Inner Authentication Method MSCHAPv2

4.2 Analysis on Vulnerabilities in EAP-TTLS Using Inner Authentication Method PAP

PAP uses the minimum form of authentication which requires supplicant to provide their username and password which is compared to a table of ID-Password pairs stored in the server. Username and password provided by the supplicant during authentication were sent unencrypted. This vulnerability was proven to be true in the experiment conducted.

4.2.1 Cracking EAP-TTLS Using Inner Authentication Method PAP

The attacker was able to sniff or eavesdrop the network authentication between the supplicant and AS. Its username (SecurityTest) and password (w0rdP@$$) were successfully obtained as shown in figure 18.

The significance of the outcome indicates PAP is not a strong authentication method, an attacker can easily obtain username and password of the supplicant due to the unencrypted user credentials transmitted over the network. It does not matter the size, length and complexity of the username and password. Although PAP is vulnerable, there are certain instances where it could be considered useful. In situations where incompatibility issues exist between varied vendors in implementation of CHAP, PAP can be implemented since many networking OS remote servers supports it.

Figure 18: Captured Unencrypted Username and Password Using Inner Authentication Method PAP

5. CONCLUSION

The use of penetration testing can help network administrators to improve their network security by assessing and identifying vulnerabilities as a result find solutions to them before an attacker takes the lead.

PEAP and EAP-TTLS does not validate supplicants with both client-side and server -side digital certificate. They only validate with server-side digital certificates which are vulnerable to receiving fake digital certificates.

MSCHAPv2 is based on a challenge-response mechanism which is sent in plaintext. MSCHAPv2 is vulnerable to dictionary attack; if the password exists in the dictionary it will be successfully cracked.

PAP Inner Authentication Method uses a two-way handshake to authenticate supplicants. PAP requires supplicants to provide their username and password to the AS. The username and password traverse over the network are sent in plaintext unencrypted which makes it easier for an attacker to capture supplicant credentials. PAP is vulnerable to eavesdropping and impersonation. PAP does not require any dictionary to crack since suppliant credentials are not encrypted over the network.

EAP-TLS uses both client-side and server-side digital certificates which makes it difficult to be cracked, highly recommended for organizations.

Configurations and settings for PEAP or EAP-TTLS, network administrators must ensure that certificate validation is turned on, the permitted certifying authorities are checked and functionalities that enables users to accept new certifying authorities, certificates, and Radius Servers are turned off.

Currently, EAP-TLS is the only secured authentication method recommended to be configured by network administrators if resources are available. Both client-side and server-side certificates is a mandatory in EAP-TLS. Vulnerabilities associated with this type of EAP is yet to be discovered. It is recommended for large enterprises with sensitive information. Depending on the EAP authentication method implemented, digital certificates should be frequently updated.

6. REFERENCES

[1] Kissi, M. K. and Asante, M., (2020). Penetration Testing of IEEE 802.11 Encryption Protocols Using Kali Linux Hacking Tools. International Journal of Computer Applications (0975 – 8887) Volume 176 – No. 32, June 2020.

[2] Kothaluru, R. T. and Youshah, M. S. M., (2012). Evaluation of EAP Authentication Methods in Wired and Wireless Networks. School of Computing, Blekinge Institute of Technology, Sweden.

[3] Kachhara, S. and Kumar, K. A., (2018). Implementation of IEEE 802.1x Port-based Authentication Mechanism for Ethernet. International Journal of Computer Trends and Technology (IJCTT) – Volume 64 Number 1 – October 2018.

[4] Kumar, U., Kumar, P. and Gambhir, S., (2014). Analysis and Literature Review of IEEE 802.1x (Authentication) Protocols. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-3, Issue-5, June 2014

[5] Thomas, T. and Stoddard, D., (2011). Network Security First-Step, Second Edition, Indianapolis, Cisco press, ch. 6, pp. 169-192.

[6] IEEE Computer Society, (2012). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard, IEEE, March 2012.

[7] Baheti, Akshay, (2015). "Extensible Authentication Protocol Vulnerabilities and Improvements". Retrieved from http://scholarworks.sjsu.edu/etd_projects/425 (Accessed on November 20, 2016)

[8] Robyns P., (2014). Wireless Network Privacy. Hasselt University

[9] Strand L., (2004). Logical port entities diagram. Retrieved from http://www.tldp.org/HOWTO/8021X-HOWTO/intro.html. (Accessed on July 6, 2018)

[10] Wikimedia (2010). 802.1X involved protocols diagram.

Retrieved from http://upload.wikimedia.org/wikipedia/commons/1/1f/802.1X_wired_protocols.png. (Accessed on July 7, 2018).

[11] Aboba B., Levkowetz E. H., Vollbrecht J., Carlson J., and Blunk L., "Extensible Authentication Protocol (EAP)". RFC 3748, IETF, June 2004.

[12] Blunk L. and Vollbrecht J., (1998). PPP Extensible Authentication Protocol (EAP). RFC 2284, IETF, March 1998.

[13] Memon A. Q., Raza A. H. and Iqbal S., (2010). WLAN Security. Halmstad University School of Information Science, Computer and Electrical Engineering. Technical report, IDE1013, April 2010.

[14] Aboba B., and Calhoun P., (2003). RADIUS support for EAP. RFC 3579, IETF, September 2003.

[15] Congdon P., Aboba B., Smith A., Zorn G., and Roese J., (2003). IEEE 802.1X RADIUS Usage Guidelines. RFC 3580, IETF, September 2003.

[16] Madjid N. and Mahsa N., (2005). "AAA and Network Security for Mobile Access: RADIUS, DIAMETER, EAP, PKI and IP Mobility". John Wiley & Sons Ltd. The Atrium, Southern Gate, Chichester, West Sussex P0198SQ, England. ISBN-13 978-0-470-01194-2. January 2005.

[17] Ramachandran, V. (2011), BackTrack 5 Wireless Penetration Testing, Master Bleeding Edge Wireless Testing Techniques with BackTrack 5: Packt Publishing, Birmingham UK

[18] Macnally C., (2001). Cisco LEAP protocol description. Protocol description, IETF, September 2001.

[19] Gast M., (2004). TTLS and PEAP Comparison. InteropNet Labs. Retrieved from www.opus1.com/www/whitepapers/ ttlsandpeap.pdf (Accessed on December 29, 2017).

[20] Interlink Networks, Inc (2003). Configuring PEAP and TTLS in the Interlink Networks RAD-Series and Secure.XS RADIUS Servers.

[21] Hoeper K. and Chen L., (2009). Recommendation for EAP Methods Used in Wireless Network Access Authentication. Computer Security Division Information Technology Laboratory, NIST Special Publication 800-120, September 2009.

[22] Gast M., (2004). Inner Authentication Methods, InteropNet Labs. Retrieved from www.opus1.com/www/whitepapers/8021xinnerauthmethods.pdf. (Accessed on December 29, 2017).

[23] Schneier B., Mudge, and Wagner D., (1999). Cryptanalysis of Microsoft's PPTP Authentication Extensions. CQRE '99, October 1999.

[24] Eisinger J., (2001). Exploiting known security holes in Microsoft's PPTP Authentication Extensions (MS-CHAPv2). University of Freiburg

[25] Rahbar, A. (2012). Weaknesses in MS-CHAPv2 authentication. Retrieved from https://blogs.technet.microsoft.com/srd/2012/08/20/weaknesses-in-ms-chapv2-authentication (Accessed on July 16, 2018)

[26] Ghering M., (2016). Evil Twin vulnerabilities in Wi-Fi networks. Radboud University

[27] Rebane R., (2016). Security of passwords in Eduroam. University of Tartu, Institute of Computer Science, Computer Science Curriculum.

[28] Marlinspike M., (2012). Divide and Conquer: Cracking MS-CHAPv2 with a 100via Internet Archive: Wayback Machine, July 2012. Retrieved from https://web.archive.org/web/20160316174007/https://www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2/. (Accessed on September 7, 2018).

[29] Lamotte W., Robyns P., Bonné B., Quax P., (2014). Exploiting WPA2 Enterprise Vendor Implementation Weaknesses through Challenge Response Oracles. Hasselt University.