IMPLEMENTASI DAN ANALISIS AUTENTIKASI JARINGAN WIRELESS MENGGUNAKAN METODE *EXTENSIBLE AUTHENTICATION PROTOCOL – TRANSPORT LAYER SECURITY* (EAP-TLS)

Evans Batrinixon Lumban Gaol[1], Cokorda Rai Adi Pramartha, S.T., MMSI.[2]
Program Studi Teknik Informatika, Jurusan Ilmu Komputer,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Email: evans.batrinixon@cs.unud.ac.id[1], cokorda@cs.unud.ac.id[2]

ABSTRAK

Layanan keamanan secara umum terdiri dari dua proses yaitu autentikasi dan enkripsi. Autentikasi jaringan dengan *username* dan *password* (*captive portal*) saat ini sangat rentan terhadap serangan. Data berupa *username* dan *password* dapat dicuri untuk memperoleh akses jaringan secara ilegal.

Pada penelitian ini, sistem autentikasi dengan *Extensible Authentication Protocol - Transport Layer Security* (EAP-TLS) diterapkan untuk mengatasi kelemahan sistem autentikasi yang menggunakan *username* dan *password.* EAP-TLS membutuhkan *Public Key Infrastructure* (PKI) untuk meningkatkan level keamanan dengan adanya kunci publik dan kunci privat memanfaatkan sertifikat digital.

Dengan menggunakan *Advanced Encryption Algorithm* (AES) 128-bit dan Diffie-Hellman RSA untuk pertukaran kunci, komunikasi yang aman akan tercipta. Sistem autentikasi EAP-TLS dapat mengatasi serangan terhadap sistem autentikasi *captive portal* yaitu sniffing *password* dan *spoofing* MAC *address*.

Kata Kunci*: EAP-TLS, Public Key Infrastructure, Sertifikat Digital, Autentikasi, Wireless-LAN*

ABSTRACT

In general, the security services consist of two processes; they are authentication and encryption. The network authentication using the username and password (captive portal) is currently sensitive to attacks. The data in the forms of username and password may be stolen in order to acquire illegal access.

In this present study, the system of authentication with the Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) was applied to overcoming the weakness of the authentication system using the username and password. EAP-TLS needs Public Key Infrastructure (PKI) to improve the level of security as a consequence of the availability of the public and private keys using digital certificate.

Safe communication will be created if the Advanced Encryption Algorithm (AES) 128-bit and Deffie-Hellman RSA are used for the key exchange. The system of the authentication of EAP-TLS can overcome any attack against the captive portal authentication, that is, sniffing password and spoofing MAC address.

Keywords: EAP-TLS, Public Key Infrastructure, Digital Certificate, Authentication, Wireless-LAN

1. Pendahuluan

Celah keamanan yang dapat menjadi permasalahan pada jaringan wireless adalah sistem koneksi dan autentikasi bagi pengguna. Autentikasi tersebut dibutuhkan bagi pengguna jaringan agar dapat terhubung dengan jaringan wireless secara legal (Sukmaaji dan Rianto, 2008).

Adanya celah keamanan pada suatu jaringan wireless dapat dimanfaatkan oleh siapa saja untuk melakukan hal ilegal seperti menyadap segala informasi yang ada melalui jaringan. Apabila hal ini terjadi maka informasi yang diperoleh dapat saja digunakan untuk hal-hal yang merugikan.

Untuk meningkatkan sistem keamanan jaringan wireless digunakan koneksi wireless yang lebih aman dengan adanya autentikasi. Autentikasi memungkinkan adanya user yang boleh mengakses jaringan dan user yang tidak diizinkan mengakses jaringan (Sukmaaji dan Rianto, 2008). Proses autentikasi yang sering dilakukan adalah dengan cara verifikasi *username* dan *password* dari user yang akan mengakses jaringan, Cara ini tidak menjamin bahwa user

yang akan mengakses jaringan adalah user yang diinginkan karena *username* dan *password* yang dimiliki dapat saja diketahui oleh orang lain.

Oleh karena itu, perlu dikembangkan lagi mekanisme keamanan jaringan yang secara spesifik digunakan pada wireless. Salah satu mekanisme autentikasi yang dapat digunakan adalah *Extensible Authentication Protocol* (EAP) dengan metode *Extensible Authentication Protocol* (EAP) – Transport Layer Security (EAP-TLS).

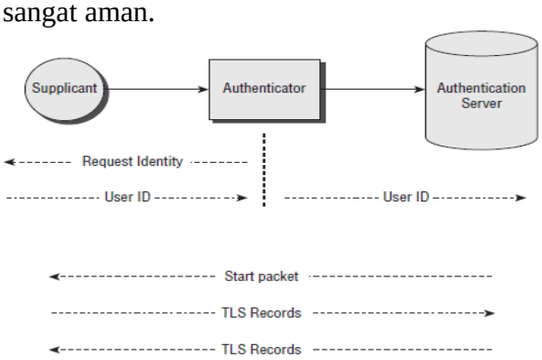
EAP-TLS adalah salah satu metode autentikasi EAP berbasis sertifikat. EAP-TLS menggunakan sertifikat kunci publik untuk autentikasi dari *client* ke *server* dan server ke client (Arifin, 2008). Komponen penting dalam menerapkan EAP-TLS adalah *Public Key Infrastructure* (PKI). *Public Key Infrastructure* merupakan sebuah sistem dari sertifikat digital yang memeriksa dan mengotentikasi validitas dari setiap entitas yang berpartisipasi dalam mengamankan komunikasi melalui penggunaan kriptografi public key (Arifin, 2008).

2. Tinjauan Pustaka

2.1 EAP dengan Transport Layer Security (EAP-TLS)

EAP dengan Transport Layer Security (TLS) mengharuskan adanya autentikasi timbal-balik dimana baik *supplicant* dan *server* autentikasi saling membuktikan identitas mereka satu sama lain. EAP-TLS membuat penggunaan kriptografi kunci publik untuk tujuan autentikasi, yang mana melibatkan *smart card* atau sertifikat digital.

Kemampuan metode EAP-TLS menggunakan sertifikat membuat EAP-TLS cocok untuk mengontrol akses pada lingkungan jaringan *wireless*, dimana *client*, yang sebelumya tidak memiliki koneksi yang dapat dipercaya dengan suatu jaringan, dapat mengautentikasi diri dan selanjutnya mempertukarkan sertifikat untuk membangun saluran komunikasi yang aman dengan jaringan (Nakhjiri, 2005). Komunikasi antara *supplicant* dan *server* autentikasi direalisasikan via *tunnel* TLS yang dienkripsi. Hal ini membuat EAP-TLS sangat aman.



Gambar 1. Proses Autentikasi EAP-TLS

2.2 Sertifikat dan Public Key Infrastructure (PKI)

Mekanisme autentikasi pada jaringan *wireless* berbasis pada 802.1X yang menggunakan EAP-TLS memerlukan sertifikat. Pada autentikasi *wireless* berbasis EAP-TLS, *client* dan *server* saling mempertukarkan sertifikat.

Pada saat berlangsungnya pertukaran sertifikat, pengamanan informasi sangat penting untuk dilakukan. Salah satu teknik pengamanan informasi yang dapat dilakukan adalah teknik enkripsi. Pada EAP-TLS enkripsi yang digunakan untuk mengamankan sertifikat adalah enkripsi *public key* yang menggunakan dua kunci berbeda untuk setiap bagian yang saling berkomunikasi.

Menurut Arifin (2008), teknologi enkripsi *public key* juga mengizinkan untuk memasangkan *digital signature* pada sebuah pesan. Untuk membuat sebuah *digital signature*, pengirim menghitung *hash* dari sebuah pesan. *Hash* adalah sebuah nilai yang mewakili sebuah pesan. Pengirim kemudian mengenkripsi *hash* menggunakan *private key*. *Hash* yang sudah terenkripsi merupakan *digital signature* dari sebuah pesan yang akan dikirim. Ketika pesan dan *digital signature* diterima, penerima akan melakukan penghitungan nilai *hash* untuk sebuah pesan. Penerima menggunakan *public key* yang berhubungan dengan pengirim untuk mendekripsi *digital signature* dan memverifikasi bahwa *hash* yang digunakan sama dengan *hash* yang dihasilkan. Jika mereka sama, artinya selama pengiriman pesan tidak mengalami perubahan.

Untuk mengamankan integritas *public key*, *public key* dipublikasikan sebagai bagian dari sebuah sertifikat. Sertifikat (*digital certificate* atau *public key certificate*) merupakan sebuah struktur data yang berisi sebuah *digital signature* dari CA (*Certificate Authority*: sebuah entitas yang dapat dipercaya oleh *user*).

Sertifikat merupakan sebuah *statement digital sign* yang mengaitkan antara nilai dari sebuah *public key* dengan identitas *user* atau perangkat yang memiliki *private key* yang saling berhubungan. Sebuah sertifikat dibentuk dari serangkaian *field* yang berisi informasi, informasi tersebut diperlukan untuk:

1. Mengidentifikasi subjek dari sebuah sertifikat dan yang berhubungan dengan *public key*.
2. Mengidentifikasi pemberi sertifikat.
3. Memverifikasi keabsahan sertifikat.

Adapun *Public key infrastucture* (PKI) adalah sebuah sistem dari sertifikat digital dan CA yang memeriksa dan mengotentikasi

validitas dari setiap entitas yang berpartisipasi dalam mengamankan komunikasi melalui penggunaan kriptografi kunci publik (Arifin, 2008).

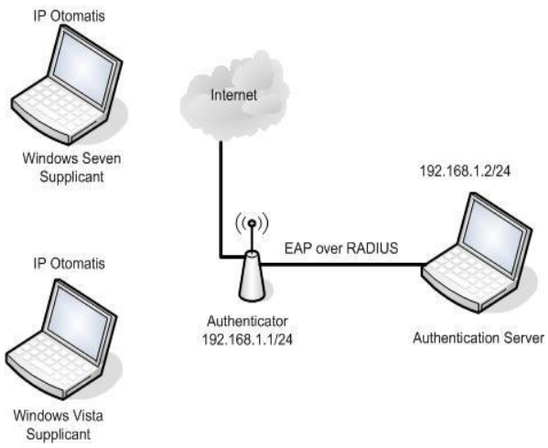
Ketika sebuah sertifikat diberikan ke sebuah entitas dan menjadi pengenal bagi pemegang sertifikat, hal tersebut hanya berguna jika entitas yang bersangkutan mempercayai *Certificate Authority* (CA) sebagai pemberi sertifikat.

Ketika sebuah entitas mempercayai sebuah CA, ia harus percaya bahwa CA menetapkan kebijakan yang tepat untuk memeriksa permintaan sertifikat dan akan menolak permintaan sertifikat jika entitas tersebut tidak memenuhi kebijakan yang telah ditetapkan oleh CA.

3. Tahap Desain

3.1 Desain Skema

Skema sistem yang diimplementasikan terdiri dari 3 komponen yaitu *supplicant*, *authenticator*, dan *authenticator server*. *Authenticator* memiliki alamat jaringan 192.168.1.1, sedangkan *authenticator server* memiliki alamat jaringan 192.168.1.2. Adapun *supplicant* akan menerima alamat jaringan secara otomatis melalui DHCP *server*. Skema topologi sistem yang diimplementasikan adalah sebagai berikut.

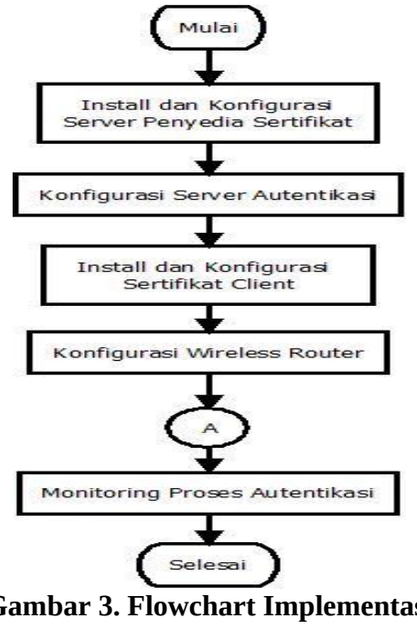


Gambar 2. Skema Sistem EAP-TLS

Sistem yang diimplementasi menyediakan autentikasi jaringan untuk mengakses jaringan. Tiap *user* yang mengakses jaringan diverifikasi oleh *server* sebelum mengakses jaringan dan *server* juga akan memverifikasi diri ke *user*. Sesuai dengan bab I, parameter identitas yang akan diverifikasi adalah sertifikat yang dimiliki oleh *supplicant* dan *server*. Jika proses verifikasi telah selesai dan sesuai dengan kebijakan yang telah ditetapkan maka *supplicant* dapat mengakses jaringan. Berikut skema kerja sistem yang dirancang:

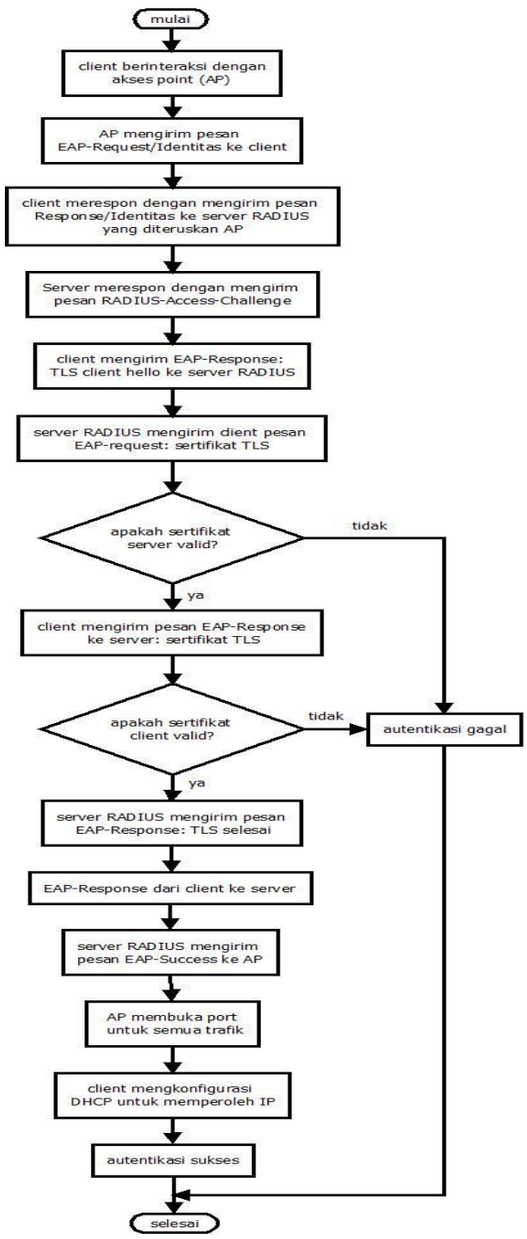
1. Proses dimulai dari asosiasi *user* terhadap akses point. Pada proses ini, *supplicant* harus terkoneksi ke jaringan. Pengguna akan diberikan sertifikat yang dikonfigurasi oleh administrator jaringan. Pada akhir proses ini, *supplicant* akan memperoleh sertifikatnya sendiri.
2. *Supplicant* telah siap untuk membangun koneksi melalui akses point dengan berkomunikasi dengan *server* autentikasi.
3. Ketika *supplicant* ingin mengakses jaringan, maka akses point yang juga berfungsi sebagai autentikator akan meminta identitas *supplicant* yaitu sertifikat. Jadi yang dilewatkan ketika proses ini berlangsung hanyalah trafik EAP saja.
4. Akses point sebagai autentikator berfungsi untuk me-*relay* paket identifikasi yang dikirim ke *server* autentikasi (RADIUS). Agar autentikator dan *server* autentikasi dapat saling berkomunikasi maka *shared-secret* pada autentikator terlebih dahulu harus didefinisikan pada *server* autentikasi. Jadi *shared-secret* yang ada harus sama.
5. Jika *server* autentikasi menyatakan bahwa sertifikat yang dikirim oleh *supplicant* valid dan begitu juga sebaliknya, maka proses selanjutnya adalah konfigurasi protocol lainnya misalnya DHCP untuk memperoleh IP *address.*
6. Akhirnya *supplicant* dapat mengakses jaringan.

Server autentikasi yang menyediakan sertifikat adalah *server* yang bertindak sebagai *root* CA sekaligus penyedia sertifikat. Oleh karena itu, *server* autentikasi harus memiliki 3 sertifikat yaitu sertifikat *public*, sertifikat *private key server*, dan sertifikat CA *root*. Adapun *supplicant* harus memiliki juga 2 sertifikat yaitu sertifikat *public* dan sertifikat *private key client*.



Gambar 3. Flowchart Implementasi Sistem Autentikasi EAP-TLS

Pada tahap konfigurasi *server* penyedia sertifikat yaitu *server* autentikasi dilakukan konfigurasi seperti memilih metode autentikasi EAP-TLS yang digunakan serta pembuatan sertifikat untuk *server* dan *client*. Pada flowchart terdapat konektor yang disimbolkan olehhuruf A. Konektor ini merupakan proses autentikasi dari EAP-TLS secara lengkap seperti ditunjukkan pada gambar 4.



Gambar 4. Flowchart Proses Autentikasi EAP-TLS (A)

3.2 Pertukaran Kunci Pada EAP-TLS

EAP-TLS merupakan metode autentikasi yang saling mempertukarkan pesan, menyediakan proses negosiasi, dan menentukan kunci enkripsi. Proses-proses tersebut membutuhkan pertukaran kunci sertifikat antara *server* dan *client*. Sertifikat tersebut yang nantinya diharapkan akan mengautentikasi *server* maupun *client*.

Proses pertukaran sertifikat dimulai setelah *client* mengirim pesan EAP-Response: TLS *client hello* ke *server*. Setelah itu, *server* akan mengirimkan sertifikat (*public key*) *server* ke *client*. Sertifikat *server* akan diverifikasi oleh *client*, jika sertifikat benar maka *client* lalu mengirimkan sertifikat *public key client* ke *server*. *Client* lalu akan menghasilkan kunci simetrik yang akan dikirimkan ke *server*. Pesan dienkripsi dengan kunci publik *server*. Pesan yang telah ditandatangani dengan *private key client* akan dikirimkan ke *server* untuk membuktikan identitas *client*. Sampai di bagian ini, verifikasi proses autentikasi dan pertukaran kunci telah berjalan dengan baik oleh *client*. Setelah menyelesaikan negosiasi *cipher* selesai, *server* juga melakukan verifikasi proses autentikasi dan menyelesaikan pertukaran kunci.

4. Implementasi dan Pengujian

Setelah melakukan konfigurasi installasi dan konfigurasi server, tahap terpenting dari implementasi EAP-TLS adalah pada saat pembuatan sertifikat digital.

4.1 Pembuatan Sertifikat Digital

Untuk membuat sertfikat digital yang dibutuhkan maka openSSL perlu dikonfigurasi sesuai dengan kebutuhan.

1. Pembuatan Sertifikat Digital Root CA
   Konfigurasi dilakukan pada *server* yang berjalan pada sistem operasi Ubuntu 10.10. Konfigurasi file openssl.cnf yang berada pada direktori /etc/ssl mencakup:
   - Default_crl_days = 365; artinya waktu berlaku sertifikat selama 1 tahun
   - Default_bits = 1024; artinya besar bits untuk enkripsi

   Untuk membuat sertifikat *root* CA diperlukan file CA.root.sh yang berada di direktori /etc/freeradius/certs. File CA.root.sh berisi *script* yang akan dieksekusi untuk permohonan dan *signing* sertifikat.

   ```
   root@server:/etc/freeradius/certs# sh
   CA.root.sh ilkomcert
   ```

   Proses permintaan untuk pembuatan sertifikat akan membutuhkan informasi berupa *pass phrase* untuk *private key* dan DN. Eksekusi *script* akan menghasilkan tiga file sertifikat yaitu root.der, root.p12, dan root.pem. Isi dari sertifikat yang dihasilkan antara lain: nomor serial, DN, masa berlaku, jenis algoritma tanda tangan digital, dan jenis algoritma enkripsi.
2. Pembuatan Sertifikat Digital *Server*
   Tahap yang dilakukan hampir sama dengan pembuatan sertifikat root.CA. Pembuatan sertifikat untuk *server* membutuhkan file CA.*server*.sh.

   ```
   root@server:/etc/freeradius/certs# sh
   CA.server.sh serverilkom servercert
   ilkomcert
   ```

Eksekusi *script* akan menghasilkan tiga file sertifikat yaitu serverilkom.der, serverilkom.p12, dan serverilkom.pem. Sesuai *script* pada file CA.*server*.sh, "serverilkom", "servercert", dan "ilkomcert" berturut-turut adalah nama *server*, *password server*, dan *password root*.

3. Pembuatan Sertifikat Digital *Client*

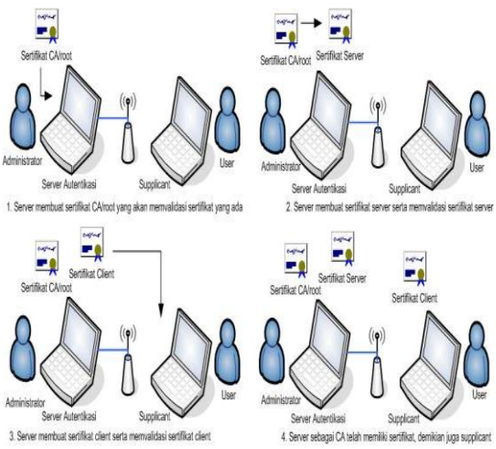
   Pembuatan sertifikat untuk *client* membutuhkan file CA.*client*.sh.

   ```
   root@server:/etc/freeradius/certs# sh
   CA.client.sh client1 client1cert ilkomcert
   ```

   Eksekusi *script* menghasilkan tiga file sertifikat yaitu client1.der, client1.p12, dan client1.pem untuk *client* berbasis sistem operasi Windows Vista. Langkah yang sama dilakukan untuk membuat sertifikat bagi *client* berbasis sistem operasi Windows Seven yaitu client1.der, client1.p12, dan client2.pem.

   ```
   root@server:/etc/freeradius/certs# sh
   CA.client.sh client2 client2cert ilkomcert
   ```

   Sertifikat yang dihasilkan ini yang digunakan oleh *client* untuk mengautentikasi diri.

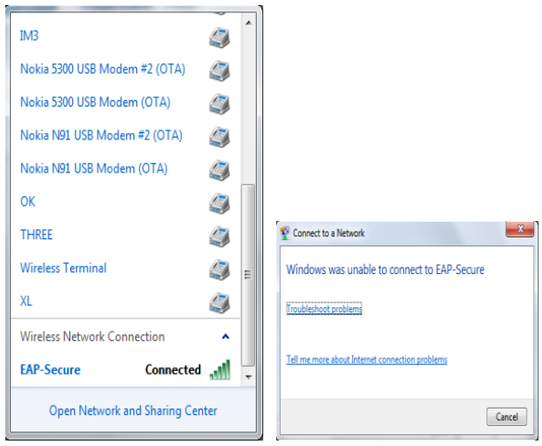


Gambar 5. Skema Pembuatan Sertifikat pada EAP-TLS

Gambar 5 memperlihatkan bagaimana tahapan pembuatan sertifikat digital pada sistem autentikasi EAP-TLS. Tahap pertama yang harus dilalui adalah administrator sistem akan membuat dan mengkonfigurasi sertifikat CA atau root yang berfungsi untuk memvalidasi sertifikat lainnya yang akan mempercayai CA. Setelah itu, *server* akan membuat lagi sertifikat yaitu sertifikat untuk *server*. Pada sistem yang telah diimplementasikan, server CA merupakan *server* autentikasi itu sendiri sehingga sertifikat *server* akan berada pada server yang sama dengan sertifikat CA/root. Agar *supplicant* dapat berkomunikasi dengan *server*, maka *user* harus memiliki sebuah sertifikat yang dipercaya oleh server CA. Oleh karena itu, *server* CA akan membuat sertifikat digital (kunci privat) *client*. Setelah sertifikat divalidasi oleh *server*, maka setiap entitas yang terlibat dalam sistem dapat berkomunikasi satu sama lain dengan adanya sertifikat digital yang valid pada masing-masing *device*.

4.2 Pengujian Sistem Autentikasi EAP-TLS

Pengujian sistem autentikasi jaringan *wireless* dengan metode EAP-TLS ini dilakukan untuk mengetahui apakah sistem yang diimplementasikan telah berjalan dengan baik atau tidak. Pengujian dilakukan dengan menggunakan laptop *client* yang telah menginstall sertifikat yang valid. Apabila *client* memiliki sertifikat yang sah maka *request* terhadap trafik jaringan akan diberikan dan *client* dapat mengakses koneksi jaringan internet. Hasil pengujian dalam hal mekanisme autentikasi membuktikan bahwa sistem autentikasi jaringan wireless yang diimplementasikan telah berjalan dengan baik baik dari sisi *server* maupun *client.*



(1) (2)

Gambar 6. Koneksi Sukses (1) dan Koneksi Gagal (2)

4.3 Pengujian Ketahanan Sistem

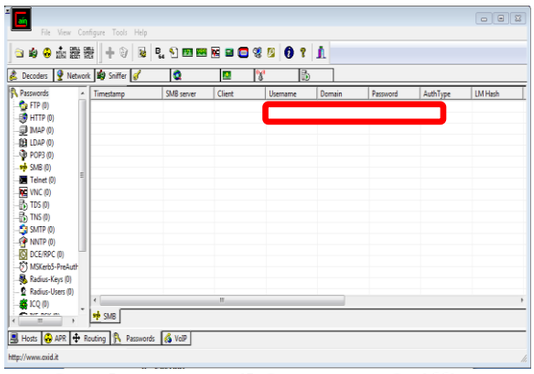
Layanan keamanan yang dimaksud adalah proses atau layanan komunikasi yang dapat menaikkan level keamanan dari sistem pemrosesan data dan transfer informasi dari pihak yang menggunakan jaringan. Layanan keamanan ini bertujuan melindungi jaringan atau sistem dari serangan-serangan yang ada.

a. Autentikasi

Untuk menguji ketahanan autentikasi jaringan EAP-TLS, penulis melakukan dua jenis serangan terhadap jaringan EAP-TLS tersebut, yaitu serangan berupa *sniffing username* dan *password* serta serangan *spoofing* IP dan MAC. Kedua jenis serangan ini dilakukan dengan menggunakan bantuan *software* Cain and Abel.

Serangan pertama berupa *sniffing username* dan *password* dilakukan sama dengan skema serangan yang telah dilakukan penulis pada saat observasi sistem autentikasi *captive portal*. Hasil serangan yang

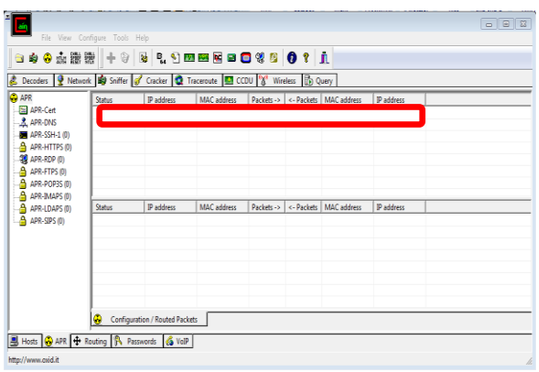
ditunjukkan pada gambar 4.6 memperlihatkan bahwa *username* dan *password* target yang diserang tidak dapat terdeteksi oleh *software* yang digunakan oleh penyerang. Hal ini disebabkan penyerang tidak memperoleh akses jaringan sehingga secara tidak langsung target yang akan diserang juga tidak akan terdeteksi oleh penyerang.



Gambar 7. Hasil Serangan Sniffing Username dan Password

Sistem autentikasi EAP-TLS melindungi pengguna jaringan yang sah dengan adanya sertifikat yang merupakan kunci untuk mengakses jaringan. Penyerang yang tidak memiliki sertifikat akan gagal pada saat tahap mengautentikasi dirinya ke jaringan. Oleh karena itu, penyerang sama sekali tidak dapat mencuri *username* dan *password* pengguna karena sistem EAP-TLS menggunakan sertifikat sebagai parameter autentikasi.

Serangan kedua yang dilakukan oleh penulis adalah *spoofing* IP dan MAC. Serangan ini dilakukan untuk mengganti IP dan MAC penyerang yang belum terautentikasi dengan IP dan MAC target yang telah terautentikasi. Hasil serangan pada gambar 7 menunjukkan bahwa serangan yang dilakukan tidak berhasil. IP dan MAC pengguna yang terhubung pada jaringan tidak dapat terdeteksi oleh penyerang. Hal ini juga disebabkan karena penyerang tidak memiliki sertifikat EAP-TLS sehingga penyerang tidak memperoleh IP dan MAC yang akan mengautentikasi si penyerang ke jaringan.

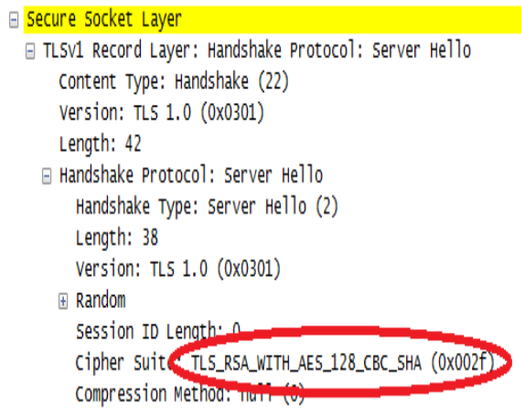


Gambar 8. Hasil Serangan Spoofing IP dan MAC

Berdasarkan dua jenis serangan yang telah dilakukan, dapat dikatakan bahwa sistem autentikasi yang menggunakan EAP-TLS yang berbasis sertifikat lebih aman karena pengguna jaringan dilindungi oleh sistem yang hanya akan mengautentikasi pengguna dengan sertifikat yang valid. Selain itu, *Service Set* ID (SSID) pada sistem autentikas EAP-TLS juga bersifat tertutup sehingga hanya yang mengetahui SSID yang valid yang dapat memliki kesempatan untuk terbuhung pada jaringan.

b. Enkripsi

Selama proses autentikasi berlangsung, *server* dan *client* akan saling mengirimkan pesan satu sama lain. Sistem yang aman akan mampu menjaga kerahasiaan atau keaslian data/pesan selama proses autentikasi tersebut. Oleh karena itu, pada sistem autentikasi EAP-TLS, komunikasi antara *server* dan *client* akan dienkripsi. Pada sistem yang telah dibuat, algoritma enkripsi yang digunakan adalah AES 128-bit sehingga kunci privat dan kunci publik tidak terlihat dalam hasil monitoring menggunakan software Wireshark.



Gambar 9. Algoritma Enkripsi pada Sistem EAP-TLS

Adapun pada sistem autentikasi berbasis *captive portal*, data penting pengguna berupa *username* dan *password* dapat terlihat karena autentikasi yang berbasis web dimana *username* dan *password* ditransmisikan melalui jaringan tanpa dienkripsi terlebih dahulu. Hal ini menyebabkan pihak-pihak lain yang tidak berhak dapat mengakses jaringan apabila mengetahui hal tersebut.

5. Kesimpulan

Mekanisme autentikasi pada sistem EAP-TLS lebih aman dibandingkan dengan sistem *captive portal* berdasarkan uji ketahanan autentikasi dengan serangan *sniffing username* dan *password* serta *spoofing* IP dan MAC. Adanya enkripsi dengan AES 128-bit juga mencegah serangan seperti

sniffing yang dapat menyerang sistem *captive portal*.

Daftar Pustaka

(1) Arifin, Zaenal. 2008. *Sistem Pengamanan Jaringan Wireless LAN Berbasis Protokol 802.1x dan Sertifikat*. Yogyakarta: Penerbit Andi.
(2) Chandra, Praphul. 2005. *Bulletproof Wireless Security*. USA: Elsevier.
(3) Cole, Dr. Eric., Dr. Ronald Krutz, James W. Conley. 2005. *Network Security Bible.* Canada: Wiley Publishing, Inc.
(4) FitzGerald, J. & Dennis, A. 2010. *Fundamental of Bussines Data Communication 1th*. Asia: Wiley.
(5) Forouzan, A. 2007. *Data Communication And Networking 4th Edition*. New York: McGraw-Hill.
(6) Karygiannis, Tom., Les Owens. 2002. *Wireless Network Security.* Gaithersburg: NIST.
(7) Kempf, James. 2008. *Wireless Internet Security.* London: Cambridge University KSC.
(8) Nakhjiri, Madjid., Mahsa Nakhjiri. 2005. *AAA and Network Security For Access*. Asia: Wiley
(9) P. Clark , Martin. 2003. *Data Network, IP, dan Internet*. Germany: Wiley.
(10) Sukmaaji, Anjik S.Kom., Rianto, S. Kom. 2008. *Jaringan Komputer*. Yogyakarta: Penerbit Andi.
(11) Zhang, Yan., Jun Zheng, Miao Ma. 2008. *Wireless Security*. New York: Information Science Reference.

[Halaman ini sengaja dikosongkan]