Jurnal Integrasi
Vol. 9 No. 1, April 2017, 16-23
e-ISSN: 2548 - 9828

Article History
Received March, 2017
Accepted April, 2017

Rancang Bangun Pengamanan FTP Server dengan Menggunakan *Secure Sockets Layer*

Molavi Arman
AMIK MDP
Manajemen Informatika
Jalan Rajawali No.14, Palembang, Indonesia
e-mail: molavi.arman@mdp.ac.id

Abstrak

Teknologi informasi yang berkembang pesat didalam kehidupan manusia, membuat kebutuhan akan sistem penyimpanan data terpusat menjadi sesuatu yang penting dalam penyimpanan arsip *digital*. Data tidak hanya disimpan dalam sebuah *personal computer desktop* tetapi media penyimpanan data terpusat menjadi alternatif dalam media penyimpanan, guna menjaga dari kehilangan data atau *backup* data.

Teknologi jaringan (*network*) komputer merupakan solusi yang dapat dimanfaatkan untuk memenuhi penyimpanan data. Jaringan komputer merupakan kumpulan beberapa komputer dan perangkat jaringan lain yang saling terhubung melalui media perantara.

FTP (*File Tranfer Protocol*) umumnya berfungsi sebagai media tukar menukar *file* atau data dalam suatu *network* yang menggunakan koneksi TCP. Protokol FTP tidak cukup aman dikarenakan pada saat autentifikasi *output* karakter berupa *plaintext* dan disaat *transfer* data tidak ada enkripsi untuk melindungi. Protokol FTP butuh penambahan keamanan, dengan menggunakan protokol TLS *(Transport Layer Security)* dan *Auth* SSL untuk mengamankan protokol FTP pada saat autentikasi dan proses *transfer* data.

Untuk melindungi FTP *server* dari kerentanan autentikasi dan pengiriman data perlu ditambahkan fitur keamanan menggunakan *Secure Sockets Layer* (SSL) untuk mengenkripsi protokol FTP pada saat autentikasi dan proses transfer data. Sertifikat SSL digunakan untuk menangani keamanan pada paket data yang ditransmisikan melalui jaringan. Ketika SSL digunakan, maka *server* atau penyedia jasa akan memberikan sertifikat publik ke klien untuk melakukan autentikasi keabsahan identitas dari *server*. Ketika sudah terautentikasi, maka koneksi antara *server* dengan klien akan dienkripsi.

Kata Kunci: Kinerja FTP, Jaringan Komputer, *Database*, PHP, TLS, SSL.

Abstract

The rapid development of information technology in human life makes the need of centralized data storage system become an important thing in digital archives storage. The data is not only stored in personal computer desktop but also protected to keep the data or data back up.

Computer network technology is a solution that can be used to save the data. Computer network is a group of computers and other network device connected to each other through media.

FTP (File Transfer Protocol) generally functions as media to transfer file or data in a network using TCP connection. FTP protocol is not secure enough since during the authentication the character of the output is in the form of plaintext and during data transfer there is no encryption to protect. FTP protocol needs additional security using TLS (Transport Layer Security) and Auth SSL to secure FTP protocol during authentication and data transfer process.

To protect FTP server from the authentication susceptibility and data transfer, security feature using Secure Socket layer (SSL) should be added to encrypt FTP protocol during authentication and data transfer. SSL certifícate is used to handle the security of data packet transmitted through the network. When SSL is used, server will give public certifícate to the client to do authentication of identity from the server. After being authenticated, the connection between server and client will be encrypted.

Keywords: FTP, Computer Network, Database, PHP, TLS, SSL.

1. Pendahuluan

Teknologi informasi yang berkembang pesat didalam kehidupan manusia, membuat kebutuhan akan sistem penyimpanan data terpusat menjadi sesuatu yang penting dalam penyimpanan arsip dan dokumen *digital*. Data tidak hanya disimpan dalam sebuah p*c desktop* atau media penyimpanan saja tetapi media penyimpanan data terpusat menjadi alternative dalam media penyimpanan, guna menjaga dari kehilangan data atau *backup* data.

Teknologi Jaringan (*network*) komputer merupakan solusi yang dapat dimanfaatkan untuk memenuhi penyimpanan data. Jaringan komputer merupakan kumpulan beberapa komputer (dan perangkat lain seperti: *printer*, *hub* dan *switch* dan sebagainya) yang saling terhubung satu sama lain melalui media perantara. Jadi dengan adanya jaringan komputer tersebut penyimpanan data terpusat dapat dilakukan dengan baik.

FTP (*File Tranfer Protocol*) umumnya berfungsi sebagai media tukar menukar *file* atau data dalam suatu *network* yang menggunakan TCP koneksi. FTP yang digunakan menggunakan berbasis *Open Source* guna menunjang tingkat stabilitas tinggi dan tidak mudah terinfeksi *virus* dan *malware*. FTP merupakan metode protokol pilihan yang paling tepat dalam penyimpanan *file/*data secara cepat dalam proses *upload* dan *download* dari komputer *server* ke klien tanpa menggunakan *flashdisk* untuk mengambil data dari komputer *server*.

Protokol FTP tidak didisain secara aman , FTP memiliki kelemahan dan kendala dalam autentikasi serta pengiriman data dari server ke klien atau sebaliknya, kelemahan inilah cukup membahanyakan jika digunakan pihak-pihak tertentu untuk melakukan penyadapan akun dan data dijaringan.[1]

Untuk melindungi FTP *server* dari kerentanan autentikasi dan pengiriman data perlu ditambahkan fitur keamanan menggunakan *Secure Sockets Layer* (SSL) untuk mengenkripsi protokol FTP pada saat autentikasi dan proses transfer data. Sertifikat SSL digunakan untuk menangani keamanan pada paket data yang ditransmisikan melalui jaringan. Ketika SSL digunakan, maka *server* atau penyedia jasa akan memberikan sertifikat publik ke klien untuk melakukan autentikasi keabsahan identitas dari *server*. Ketika sudah terautentikasi, maka koneksi antara *server* dengan klien akan dienkripsi.

Penerapan ini menggunakan *MySQL* yang difungsikan sebagai *database* untuk menyimpan *user* dan *profile* besarnya *quota* yang diberikan. Jumlah karyawan yang banyak, hal ini sangat menyulitkan jika harus menambahkan *user* sercara manual didalam sistem.

Proses memasukkan *user*, *quota* secara manual kedalam sistem dan menyangkut soal keamanan autentikasi serta transfer data, dengan penjelasan kendala diatas maka tema yang diambil berupa Racang Bangun Pengamanan FTP *Server* dengan Menggunakan *Protocol* SSL *(Secure Sockets Layer)*.

2. Dasar Teori

2.1. FTP *(File Transfer Protocol)*

File Transfer Protocol (FTP) merupakan client / server protokol yang menyediakan fasilitas untuk transfer data dalam jaringan atau dengan kata lain protokol yang digunakan untuk pertukaran file antara dua host dalam jaringan TCP/IP. Sebuah FTP server dapat di-set sebagai FTP publik sehingga setiap orang dapat mengakses data-data yang ada di server FTP dengan menggunakan login anonymous atau FTP. Selain itu, FTP juga dapat di-set agar server hanya dapat diakses oleh user tertentu saja dan tidak untuk public [2].

2.2 SSL *(Secure Sockets Layer)*

OpenSSL dan TLS *(Transport Layer Security)* merupakan protokol kembar yang digunakan untuk menangani keamanan paket data yang ditransmisikan melalui jaringan. Kedua protokol tersebut dikembangkan oleh Netscape. Ketika SSL digunakan, maka *server* atau penyedia jasa akan memberikan sertifikat publik ke klien dan melakukan autentikasi keabsahan indentitas dari *server*. Ketika sudah terautentikasi, maka koneksi antara *server* dengan klien akan dienkripsi. OpenSLL, merupakan aplikasi yang menghasilkan sertifikat SSL. Aplikasi ini akan digunakan oleh *server* untuk mengamankan koneksi tersebut [3].

2.3 PHP dan Apache

PHP adalah suatu bahasa pemrograman *web open source* yang digunakan secara luas terutama untuk mengembangkan web dan dapat disimpan dalam bentuk HTML. PHP dirilis pada tanggal 13 Juli 2004. PHP 5 dapat digunakan hampir semua sistem operasi utama, seperti Linux, varian UNIX (Mencakup HP-UX, Solaris dan OpenBSD), Microsoft Windows, Mac OSX. PHP juga mendukung hampir semua *web server*. PHP digunakan untuk memudahkan pengimputan nilai dalam bentuk *web base* [4].

Aplikasi *web server* yang cukup terkenal dan banyak digunakan adalah Apache yang tersedia untuk banyak sistem operasi. Hal ini dikarenakan sifat Apache yang dibangun dengan sistem modul sehingga kemampuan Apache dapat dikembangkan lebih jauh lagi. *Webserver* Apache yang digunakan adalah versi httpd-2.4.25.

2.4 *Database* MariaDB

Database adalah tempat menyimpan informasi. Memungkin untuk seseorang dengan mudah merekam dan kemudian mengakses sejumlah besar informasi untuk berbagai tujuan. Hampir semua jenis data dapat disimpan dalam *database*. *Database* dapat penyimpan nama dan alamat, catatan medis, laporan polisi,

transaksi penjualan, informasi tentang musik dan *video* koleksi, dan banyak lagi [5].

Untuk rancang bangun ini menggunakan MariaDB sebagai *database*, fungsi MariaDB dalam rancang bangun disini adalah menyimpan nama *user* dan besaran kapasitas *quota* yang digunakan untuk masing masing *user*.

3. Metodologi Penelitian

Teknik pengerjaan sebagai berikut:

a. Menyiapkan Kebutuhan

Kebutuhan yang disiapkan adalah yang berhubungan dengan rancang bangun protokol FTP *server*, protokol ssl, *database* MariaDB, Apache dan modul PHP.

b. Analisis Kebutuhan

Sistem FTP *server* yang akan dirancang bangun dengan autentikasi *user name*, *password* serta dikombinasikan dengan SSL dan tempat penyimpanan *hard drive* dengan *disk* kuota yang diintegrasikan dengan MariaDB.

c. Perancangan Sistem

Perancangan sistem ini dengan menerapkan FTP *server* dengan SSL dan pembatasan kuota pada tiap *user*-nya dengan MariaDB sebagai *database*. Pemilihan sistem operasi yang digunakan pada *server* dan klien serta perangkat lunak aplikasi yang akan digunakan.

d. Implementasi dan Pengujian

Rancang bangun dan pengujian FTP *server* dengan SSL guna keamanan saat autentikasi serta transfer data dan ditambah *disk* kuota pada *user* yang di integrasikan pada tampilan *web base* supaya memudahkan pengisian besarnya kapasitas pada *user*. Pengujian pada transfer data menggunakan *file* dengan format teks yang akan disadap dan berhasil dibaca dengan *tools* jaringan *wireshark*. Pada pengujian *disk* kuota dilakukan pembatasan kapasitas pada tiap *user* dan dilakukan *upload file* teks tersebut dengan menggunakan FTP klien yaitu FileZilla yang menandakan *user* tersebut berhasil melakukan *upload*.

e. Pembuatan Pelaporan

Penyusunan pelaporan mengumpulkan dokumentasi dengan mengikuti format yang baik dan benar yang telah ditetapkan. [6]

4. Analisis, Perancangan Dan Implementasi Pengujian

4.1 Analisa Kebutuhan *Hardware* Dan *Software*

Kebutuhan *hardware* dalam rancang bangun ini perangkat keras yang digunakan adalah sebagai berikut:

a. PC dengan spesifikasi *processor* AMD *Bulldozer* sebagai *server* FTP.

b. Memori 8GB.

c. *Hard Drive* 500GB.

d. *Monitor, keyboard, mouse, switch* 8 port dan kabel jaringan *UTP*.

Kebutuhan *software* agar penelitian ini berjalan semestinya dibutuhkan beberapa *software* yang mendukung diataranya sebagai berikut:

a. Komputer FTP *server* menggunakan sistem operasi *Linux.*

b. FTP *server* menggunakan aplikasi Proftpd.

c. FTP *client* menggunakan aplikasi FileZilla.

d. *Database* menggunakan MariaDB, simpan nama *user* dan besaran kuota.

e. Aplikasi *sniffing* menggunakan *wireshark.*

f. Untuk pembuatan sertifikat SSL menggunakan OpenSSL.

g. *Web server Apache* menjalankan *script php.*

h. Bahasa pemrograman menggunakan bahasa *php*.

4.2 Analisa Perancangan / Kebutuhan Sistem

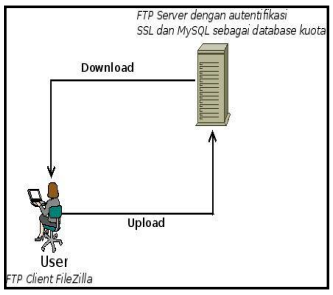
Kebutuhan sistem adalah suatu proses yang akan mengidentifikasi dan melakukan evaluasi terhadap permasalahan, dengan identifikasi dan evaluasi sehingga dibangun sistem yang sesuai. Rancang bangun pengamanan FTP server yang akan dibangun ada analisis kebutuhan, berikutnya yaitu:

a. FTP *server* menggunakan autentikasi untuk melindungi dan mengamani berkas atau *file* yang disimpan.

b. Implementasi ssl untuk melindungi proses autentifikasi, transfer data dari FTP *Client* menuju FTP *server* atau sebaliknya dan membatasi kuota.

c. Kuota yang dibatasi untuk setiap user yang terdaftar didalam *database*.

4.3 Topologi yang Akan dibangun

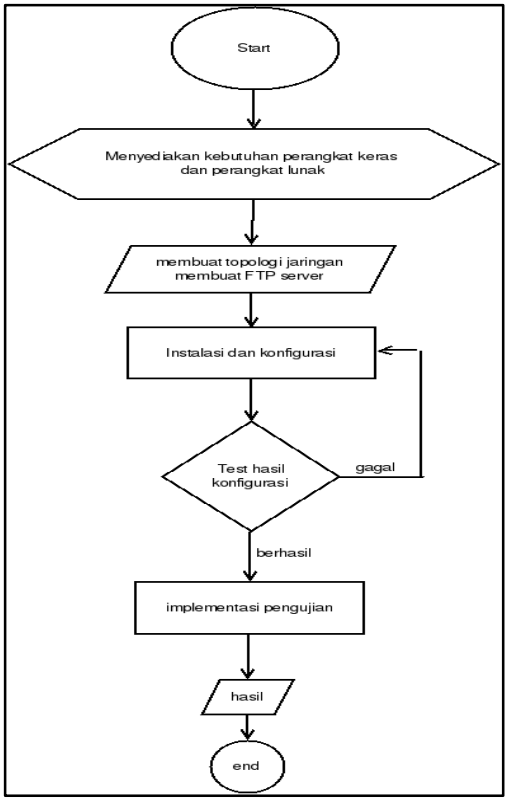


Gambar 1. Topologi yang akan dibangun

FTP *server* yang akan dibangun untuk pembuatan nama *user*, *password* dan kapasistas kuota untuk setiap *user* akan menggunakan *web base*. *User* yang akan melakukan *download* dan *upload* pada FTP *server* tanpa autentikasi *ssl user* dan password tidak akan terlindungi karena sifatnya *plaintext*. Pada FTP

server dengan autentikasi SSL nama *user, password* dan data yang berjalan didalam *network upload* maupun *download* akan terlindungi oleh sertifikat SSL.

4.4 Alur Rancang FTP *Server*



Gambar 2. Alur Rancang Bangun FTP *Server*

4.5 Perancangan FTP *Server*

Perancangan FTP *Server* ini menggunakan pengamanan *user* dan *password*. Pada saat *user name* dan *password* dibuat kemudian digunakan untuk *login* ke *FTP server* otomatis direktori tempat meletakkan berkas atau dokumen akan tercipta dengan sendirinya berbarengan pada saat *login*. *User* pada FTP *server* hanya bisa mengakses direktori *user* tersebut dan tidak bisa mengakses hak *user* lain tanpa mengetahui *user name* dan *password*nya.

4.6 Perancangan FTP *Server* dengan Secure Sockets Layer

Perancangan *File Transfer Protocol Server* yang dikombinasikan dengan Secure Sockets Layer digunakan untuk menambahkan proses pengamanan dibandingkan dengan FTP server secara *default system*. Maksud dari menambahkan pengamanan disini adalah *user name, password* dan paket data dilindungi oleh sertifkat *ssl* yang telah dikombinasikan dengan FTP *server*, sehingga terlindungi dari kegiatan *sniffing* didalam jaringan.

4.7 Perancangan Database dan Script PHP

Database menggunakan MariaDB dan *script* pemrograman php, maksudnya adalah untuk memudahkan proses pembuatan nama *user, password* dan ukuran kuota yang diberikan pada masing-masing *user* dalam bentuk *interface web base*. Komponen yang akan diisikan seperti , *user, password,* kuota dan *user id.*

4.8 Alat Uji

Untuk pengujian dan pembuktian keamanan protokol FTP dan FTPS menggunakan sebuah aplikasi yaitu wireshark atau ethereal. Wireshark adalah adalah aplikasi komputer *packet sniffer* yang bersifat *free*. Hal ini digunakan untuk mengatasi masalah jaringan, analisis, pengembangan perangkat lunak dan protokol komunikasi, dan pendidikan. Pada bulan Juni 2006, proyek ini berganti nama dari Ethereal karena masalah merek dagang. [7]

Fungsionalitas Wireshark sangat mirip dengan tcpdump, tetapi memiliki grafis *front-end* dan banyak informasi lebih lanjut menyortir dan penyaringan pilihan. Hal ini memungkinkan pengguna untuk melihat semua lalu lintas yang melewati jaringan (biasanya jaringan *Ethernet* namun dukungan yang ditambahkan untuk orang lain) dengan menempatkan antarmuka jaringan ke modus *promiscuous*. [8]

4.9 Instalasi Aplikasi MariaDB dan phpMyadmin

Database yang digunakan penulis adalah *MySQL*, aplikasi bantuan adalah *phpMyadmin* yang berjalan pada *web server* Apache untuk memudahkan manajemen pada *database* MariaDB. Langkah instalasi adalah sebagai berikut:

Tabel I

Perintah instalasi MariaDB, phpMyAdmin dan Apache

```
# apt-get install mariadb-server
  phpmyadmin apache2
```

4.10 Pemasangan Perangkat Lunak Proftpd dan Dukungan MariaDB

Sebelum melakukan instalasi aplikasi terlebih dahulu melakukan instalasi sistem operasi Ubuntu 14.10 64bit pada komputer *server*. Setelah instalasi selesai dilanjutkan dengan instalasi FTP *server* berupa aplikasi proftpd dan dukungan *database* yaitu MariaDB. Perintah instalasi adalah sebagai berikut:

Tabel II

Instalasi Proftpd dengan modul MariaDB

```
# apt-get install proftpd-mod-mysql
```

Kemudian langkah berikutnya membuat *group* FTP (*ftpgroup)* dan pengguna *(ftpuser)* bahwa semua pengguna virtual akan dipetakan menggantikan kelompok dan userid 2001 dengan nomor yang bebas di sistem.

Tabel III

Membuat *group* dan *user*

```
# groupadd -g 2001 ftpgroup
# useradd -u 2001 -s /bin/false -d /bin/null -c
  "proftpd user" -g ftpgroup ftpuser
```

4.11 Membuat *Database*

Tahap ini untuk mengintegrasikan komponen Proftpd sebagai FTP *server* dengan memasukkan kedalam *database* seperti *user*, *password*, kapasitas kuota, *userid* dan lainnya. Langkah perintah pembuatan komponen *database* proftpd sebagai berikut :

Tabel IV

Membuat *database* FTP

```
 # mysql -u root -p
CREATE DATABASE ftp;SELECT, INSERT,
UPDATE, DELETE ON
ftp.* TO 'proftpd'@'localhost'

IDENTIFIED BY 'password';GRANT
SELECT,
INSERT, UPDATE, DELETE ON ftp.* TO
 'proftpd'@'localhost.localdomain'

IDENTIFIED BY 'password';FLUSH
PRIVILEGES;
```

Tabel V

Membuat Table *ftpgroup* dan *ftpquotalimits*

```
 USE ftp;
CREATE TABLE ftpgroup (groupname
varchar(16)
NOT NULL default '',gid smallint(6) NOT
NULL
default '5500',members varchar(16) NOT
NULL default '', KEY groupname
(groupname))
ENGINE=MyISAM COMMENT='ProFTP
group table';

CREATE TABLE ftpquotalimits (name
varchar(30 ) default
NULL,quota_type enum('user','group'
,'class','all') NOT NULL default 'user'
,per_session enum('false','true') NOT
NULL
 default 'false',limit_type enum('soft','hard')
NOT NULL default 'soft',bytes_in_avail
 bigint(20) Unsigned NOT NULL
default '0',bytes_out_avail bigint(20)
unsigned NOT NULL default '0' ,
bytes_xfer_avail bigint(20) unsigned NOT
NULL default '0',files_in_avail int(10)
 unsigned NOT NULL default '0',
files_out_avail int(10) unsigned NOT
NULL default '0',files_xfer_avail int(10)
 unsigned NOT NULL default '0')
ENGINE=MyISAM;
```

Tabel VI

Membuat *ftpquotatallies* dan *ftpuser*

```
CREATE TABLE ftpquotatallies (name varchar
(30) NOT NULL default '',quota_type enum
('user','group','class','all') NOT NULL default
 'user',bytes_in_used bigint(20) unsigned NOT
 NULL default '0',bytes_out_used bigint(20)
 unsigned NOT NULL default '0',
bytes_xfer_used bigint(20) unsigned NOT
 NULL default '0',files_in_used int(10)
unsigned NOT NULL default '0',files_out_used
 int(10) unsigned NOT NULL default '0',
files_xfer_used int(10) unsigned NOT
 NULL default '0') ENGINE=MyISAM;

CREATE TABLE ftpuser (id int(10) unsigned
 NOT NULL auto_increment,userid varchar
(32) NOT NULL default '',passwd varchar
(32) NOT NULL default '',uid smallint(6)
 NOT NULL default '5500',gid smallint(6)
 NOT NULL default '5500',homedir varchar
(255) NOT NULL default '',shell varchar
(16) NOT NULL default '/sbin/nologin',
count int(11) NOT NULL default '0',accessed
datetime NOT NULL default '0000-00-00
00:00:00',modified datetime NOT NULL
default
'0000-00-00 00:00:00',PRIMARY KEY
(id),UNIQUE KEY userid (userid)) ENGINE=
MyISAM COMMENT='ProFTP user table';
quit;
```

4.13 Konfigurasi Proftpd

Melakukan *editing* pada *file* modules.conf , dengan mengaktifkan beberapa fitur [9], perintahnya sebagai berikut:

Tabel VII

Perintah edit *modules.conf*

```
# nano /etc/proftpd/modules.conf
LoadModule mod_sql.c
LoadModule mod_sql_mysql.c
LoadModule mod_quotatab_sql.
```

Dilanjutkan dengan editing file *proftpd.conf*

Tabel VIII

Mengaktifkan koneksi *MySQL* di *Proftpd*

```
DefaultRoot ~
 SQLBackend            mysql
 SQLAuthTypes          Plaintext Crypt
 SQLAuthenticate       users groups
SQLConnectInfo  ftp@localhost proftpd \
password
SQLUserInfo        ftpuser userid passwd \
 uid gid homedir shell
SQLGroupInfo      ftpgroup groupname \
gid members
SQLMinID        500
CreateHome on
SQLLog PASS updatecount
SQLNamedQuery updatecount UPDATE \
 "count=count+1, accessed=now() \
WHERE userid='%u'" ftpuser
 SQLLog  STOR,DELE modified
 SQLNamedQuery modified UPDATE\
 "modified=now() WHERE userid='%u'" \
ftpuser
 QuotaEngine on
 QuotaDirectoryTally on
 QuotaDisplayUnits Mb
 QuotaShowQuotas on
 SQLNamedQuery get-quota-limit \
SELECT "name, quota_type, per_\
session, limit_type,
 bytes_in_avail, bytes_out_avail,\
 bytes_xfer_avail, files_in_avail,
 Files_out_avail, files_xfer_avail \
FROM ftpquotalimits \
 WHERE name = '%{0}' AND  \
quota_type = '%{1}'"
 SQLNamedQuery get-quota-tally \
SELECT "name, quota_type,
 bytes_in_used, bytes_out_used, \
bytes_xfer_used,
Files_in_used, files_out_used, \
files_xfer_used FROM
 ftpquotatallies WHERE name = \
 '%{0}' AND quota_type = '%{1}'"
 SQLNamedQuery update-quota-tally \
UPDATE "bytes_in_used = bytes_in_used
 + %{0}, bytes_out_used = bytes_out_used \
+ %{1},
 bytes_xfer_used = bytes_xfer_used + \
%{2}, files_in_used = files_in_used
 + %{3}, files_out_used = files_out_used +
%{4},
 Files_xfer_used = files_xfer_used + %{5} \
WHERE name = '%{6}'
AND quota_type = '%{7}'" ftpquotatallies
 SQLNamedQuery insert-quota-tally \
INSERT "%{0}, %{1}, %{2}, %{3}, %{4},
 %{5}, %{6}, %{7}" ftpquotatallies
 QuotaLimitTable sql:/get-quota-limit
 QuotaTallyTable sql:/get-quota-tally \
/update-quota-tally/insert-quota-tally
 RootLogin off
 RequireValidShell off
```

4.14 Tahap Pemasangan SSL dan Membuat Sertifikat SSL Untuk TLS *(Transport Layer Security)*

Secure Sockets Layer adalah sebagai pelindung *otentikasi* dan *traffic* data dari *server* ke klien maupun sebaliknya. Aplikasi yang digunakanan untuk *ssl* adalah menggunakan *OpenSSL*. Perintah instalasi adalah sebagai berikut:

Tabel IX

Perintah instalasi openssl

```
# apt-get install openssl
```

Dilanjutlah dengan membuat sertifikat SSL

Tabel X

Membuat sertifikat SSL

```
# mkdir /etc/proftpd/ssl
# openssl req -new -x509 -days 365 -nodes –out
\
  /etc/proftpd/ssl/proftpd.cert.pem \
  -keyout /etc/proftpd/ssl/proftpd.key.pem
# chmod 600 /etc/proftpd/ssl/proftpd.*
```

Kemudian adalah mengaktifkan TLS *(Transport Layer Security)* dengan mengedit *file /etc/proftpd/proftpd.conf*

Tabel XI

Mengaktifkan TLS pada Proftpd

```
# nano /etc/proftpd/proftpd.conf
[...]
#
# This is used for FTPS connections
#
Include /etc/proftpd/tls.conf
[...]
```

Dilanjutkan dengan mengedit file /etc/proftpd/tls.conf

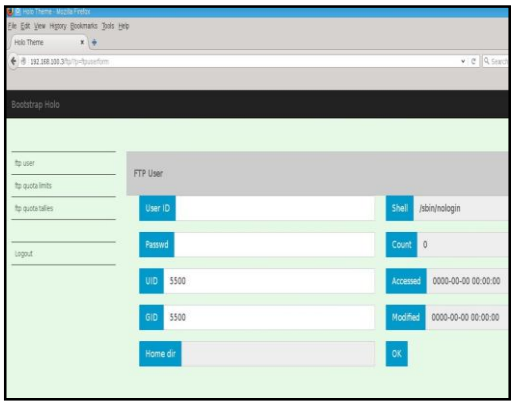
Tabel XII

tls.conf

```
<IfModule mod_tls.c>
TLSEngine               on
TLSLog                  /var/log/proftpd/tls.log
TLSProtocol TLSv1.2
TLSCipherSuite AES128+EECDH:AES128+ \
EDH
TLSOptions              NoCertRequest\
AllowClientRenegotiations
TLSRSACertificateFile     /etc/proftpd/ \
ssl/proftpd.cert.pem
TLSRSACertificateKeyFile  /etc/proftpd/ \
ssl/proftpd.key.pem
TLSVerifyClient         off
TLSRequired             on
RequireValidShell       no
</IfModule>
```

4.15 Tahap Meletakkan *Script* PHP

Script program php yang telah dibuat dimasukkan kedalam *web server* yang telah disediakan, *default /var/www/html*, kemudian tampilan *interface* sebagai berikut :

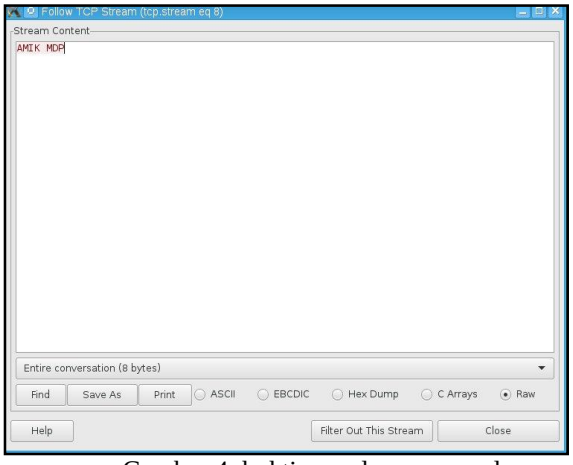


Gambar 3: *Interface* FTP *user*

4.16 Pengujian

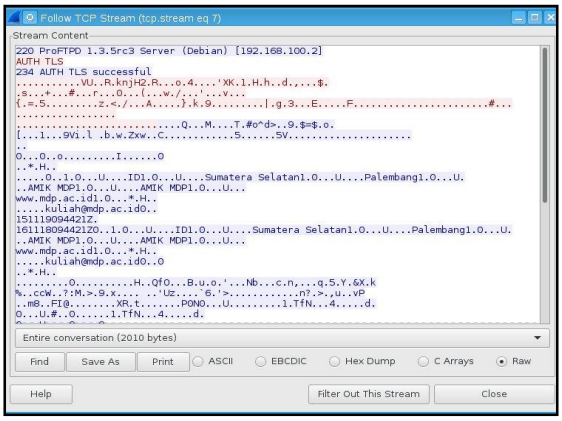
Maksud dari pengujian adalah melindungi proses autentikasi FTP *server* dengan SSL. Hal yang diuji adalah proses autentikasi FTP *server* tanpa SSL dan FTP *server* dengan SSL. Sebagai acuan keberhasilan adalah jika *username* dan *password* yang dikirim tidak bisa disadap oleh aplikasi *wireshark* dan sebagi acuan gagal bilamana *username* dan *password* mampu disadap oleh aplikasi *wireshark*.

Aplikasi FileZila melakukan proses *login* kedalam FTP *server* dengan *user name* arman , *password* 123456 dan nama file dengan nama berkas.txt isi *file* AMIK MDP terlihat pada gambar 4.



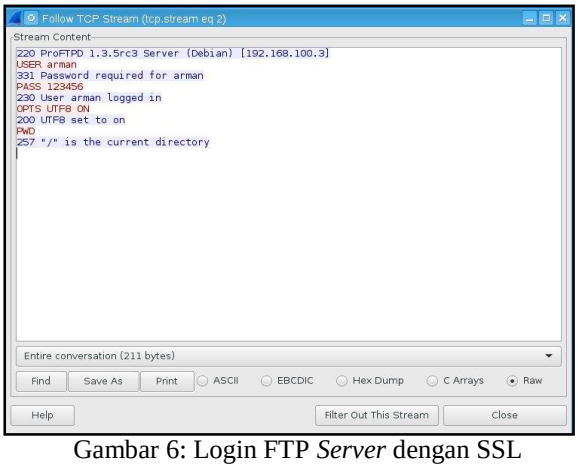
Gambar 4: bukti user dan password

Isi *file* berkas.txt yang ditangkap oleh aplikasi wireshark terlihat pada gambar 5.



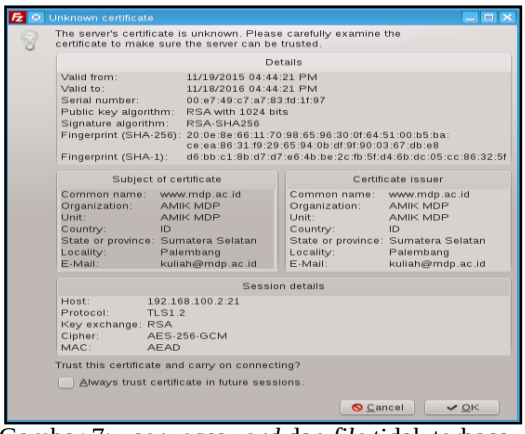
Gambar 5: Bukti isi *file* berkas.txt

FileZilla melakukan proses login kedalam FTP *server* dengan sertifikat SSL terlihat pada gambar 6.



Gambar 6: Login FTP *Server* dengan SSL

Pada FTP server dengan sertifikat SSL terlihat pada gambar xx bahwa *user*, *password* dan file yang di*upload* tidak terbaca pada aplikasi wireshark.



Gambar 7: *user*, *password* dan *file* tidak terbaca

5. Hasil Penelitian

Hasil penelitian menunjukkan bahwa:

a. Pada protokol FTP terlihat tidak terdapat pengamanan saat dilakukan penyadapan *login username* dan *password* , sehingga dapat diketahui *username* dan *password*
b. Pada protokol FTP terlihat pada saat penyadapan transfer data dari *client* menuju *server* FTP dengan terlihat isi *file* tersebut.
c. Pada protokol FTPS cukup aman pada saat *login username* dan *password* ditunjukkan pada saat penyadapan tidak terlihat *username* dan *password*
d. Pada prookol FTPS cukup aman pada saat transfer data dari *client* menuju *server*, ditujukkan tidak terlihat isi dari file yang dikirim.

6. Kesimpulan dan Saran

6.1 Kesimpulan

Kesimpulan yang dapat diambil dalam rancang bangun *FTP server* ini adalah sebagai berikut:

a. Dalam mengamankan FTP *server* dalam autentikasi yang masih standar *(plaintext*) adalah dengan menggunakan protokol ssl
b. FTP *Server* dalam transfer *file* tidak aman dengan terbacanya isi *file* berkas.txt.
c. Dalam perbandingan FTP *server* tanpa sertifikat ssl dan FTP *server* dengan sertifikat ssl bahwa menunjukkan FTP *server* tanpa sertifikat ssl tidaklah aman.
d. Kuota yang sudah dibatasi tidak akan bisa melebihi *limit* yang telah ditentukan.

6.2 Saran

Saran dalam pengembangan kedepan adalah sebagai berikut:

a. Rancang bangun ini dapat dikembangkan dengan mengintegrasikan kedalam alamat DNS, sehingga mudah dalam penamaan tidak perlu lagi mengigat IP *Address* FTP *server.*
b. Rancang bangun ini bisa dikembangkan ke akses internet dengan menggunakan IP *Address Public* serta mengintegrasikan kedalam infrastruktur DMZ.

DAFTAR PUSTAKA

[1] A.T. Sonale, S.S Matsagar, FTP Security using face recognition & Dynamic password, IOSR Journal of Computer Engineering, Second International Conference on Emerging Trends in Engineering (SICETE), Vol.1,PP:58-61. India.2013

[2] Askari Azikin., 2011, Debian Gnu / Linux. Informatika Bandung. Bandung.

[3] Imam Cartealy., 2013, Linux Networking. Jasakom. Jakarta.

[4] Wahana Komputer, 2006, Menguasai Pemrograman Web Dengan PHP 5. Andi Offset. Yogyakarta.

[5] Timothy Boronczyk., 2009, Beginning PHP6, Apache, MySQL Web Development. Willey Publishing. Indianapolis.

[6] Muhammad Martin, Ruswanda, Prajna Deshanta Ibnugraha, Tafta Zani., Implementasi FTP Server Dengan Secure Sockets Layer Dan Secure Shell Untuk Keamanan Transfer Data. Politeknik Telkom Bandung. Bandung. 2011.

[7] Ahmad Fali Oklilas, Budi Irawan, Implementasi FTP Server dengan Metode Transfer Layer Security untuk Keamanan Transfer Data Menggunakan CentOS 5.8. Portal Garuda, Vol 9, No 2 2014.

[8] Laura Chappell, Wireshark Network Analysis the Official Wireshark Certified Network Analyst Study Guide 2nd, Chappel Univerity, 2012.

[9] Srijan., 2015, Virtual Hosting With Proftpd and MySQL. [Online]. Tersedia: https://www.howtoforge.com/virtual-hosting-with-proftpd-and-mysql-incl-quota-on-ubuntu-14.04-lts. tanggal akses 10 Agustus 2015.