Analisis Penggunaan Kriptografi dalam Enkripsi Sistem *Password* Linux dan Studi Kasus Dekripsi *Password User* Linux

Hary Fernando / 13505113

Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
Jalan Ganesa 10 Bandung 40132
e-mail: if15113@students.if.itb.ac.id

ABSTRAK

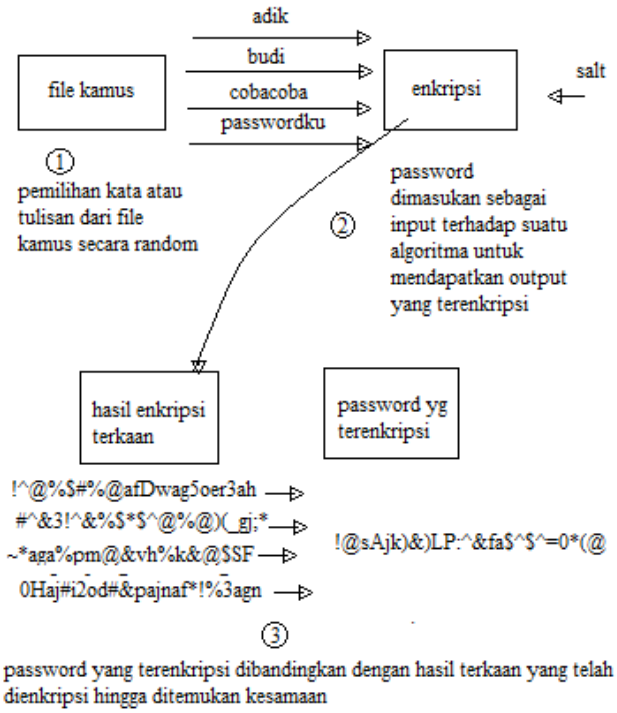
Keamanan sistem operasi sangat penting untuk menjaga semua hal yang berada di dalam sistem tersebut terbebas dari gangguan yang tidak diinginkan, baik gangguan dari dalam sistem sendiri maupun dari luar sistem. Berbagai teknik keamanan yang berkaitan dengan kriptografi yang dapat digunakan antara lain adalah proteksi sistem dengan *password* (sandi lewat), enkripsi suatu file, enkripsi seluruh partisi dari harddisk sistem, dan enkripsi dalam pengiriman pesan ke luar sistem serta serangkain teknik kriptografi lainnya.
Salah satu hal penting dalam keamanan sistem operasi adalah penggunaan *password*. *Password* yang terdapat dalam berbagai sistem operasi umumnya dienkripsi dengan algoritma yang berbeda-beda. Namun tetap saja ada pihak yang ingin melakukan dekripsi terhadap *password user* untuk tujuan yang kurang baik. Semakin maju ilmu pengetahun dan teknologi, maka teknik enkripsi yang digunakan juga semakin canggih dan sulit untuk didekripsi. Pada makalah ini penulis mencoba untuk membahas bagaimana enkripsi kriptografi bekerja pada sistem penyimpanan data *user* dan *passwrd* pada sistem Linux serta bagaimana langkah studi kasus mendekripsi dan mengetahui *user* dan *password* dari suatu sistem Linux.

Kata kunci: enkripsi *password*, *cracking*, DES, *password*, *brute-force*, John The Ripper.

1. PENDAHULUAN

Pada kebanyakan sistem UNIX (termasuk sistem Linux) generasi awal, *password* disimpan dengan suatu teknik enkripsi dengan menggunakan program `crypt` yang menggunakan algoritma yang disebut DES *(Data Encryption Standard)*[1]. Penyimpanan data *user* dan *password user* Linux ini umumnya terdapat di dalam file `/etc/passwd` serta di dalam file `/etc/shadow`. Namun seiring dengan kemajuan dan perkembangan teknologi, serta karena algoritma DES dipandang lemah dan mudah didekripsi maka teknik enkripsi yang digunakanpun semakin baik dan tahan terhadap serangan pihak yang tidak diinginkan yang mencoba mendapatkan *password user* dengan *brute force*.

Secara umum proses mendapatkan *password* dengan teknik *burte force* dapat dilihat pada Gambar 1.



Gambar 1. Teknik Brute Force untuk Mendapatkan *Password*

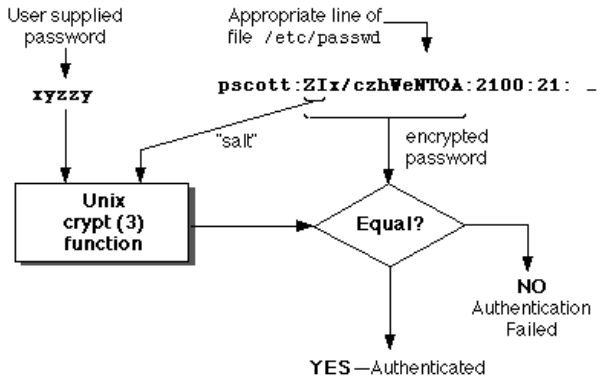
Tujuan dari makalah ini adalah mengetahui penerapan algoritma DES pada enkripsi *password* dan keamanaan *user* Linux, mengetahui bagaimana sistem *password* di Linux bekerja serta mencoba melakukan dekripsi terhadap sistem Linux.

Batasan masalah pada makalah ini adalah sistem operasi yang digunakan adalah Linux yang mendukung enkripsi dengan DES, jadi pada makalah ini tidak akan dibahas penggunaan MD5 dan varian SHA yang saat ini mendominasi enkripsi *user* dan *password* di Linux terbaru.

2. LOGIN PADA LINUX

Saat seorang *user* login ke suatu sistem Linux, maka sistem akan memanggil program `/bin/login` dan program ini akan meminta masukan *user* dan *password*. Setelah *user* memasukan *user* dan *password* dan menekan enter, kemudian masukan yang diberikan oleh *user* tersebut akan dienkripsi dengan teknik yang sama pada saat membuat passwod pertama kali dan hasil enkripsi dari masukan *user* dibandingkan dengan data yang sudah ada pada sistem, dalam hal ini file yang ada di dalam file `/etc/passwd` dan `/etc/shadow`. Jika cocok, maka berarti *password*nya sama dan *user* tersebut dibolehkan masuk sistem tersebut. Hal ini diilustrasikan dalam Gambar 2.

.



Gambar 2. Skema Login *User* Linux

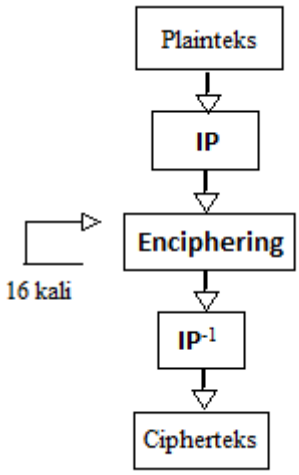
3. DES

DES merupkan singkatan dari *Data Encryption Standard* yang merupakan contoh kriptografi kunci-simetri dari jenis block cipher. DES dikembangkan di IBM pada tahun 1972. Algoritma ini berdasarkan pada algoritma Lucifer yang dibuat oleh Horst Feistel [9].

DES beroperasi pada ukuran blok 64 bit. Panjang kunci ekternalnya adalah 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai sedangkan 8 bit paritas tidak digunakan. Setiap blok (plainteks atau cipherteks) dienkripsi dalam 16 putaran. Setiap putaran menggunakan kunci internal berbeda. Kunci internal sepanjang 56-bit dibangkitkan dari kunci eksternal Setiap blok mengalami permutasi awal (IP), 16 putaran enciphering, dan inversi permutasi awal (IP-1). Skema dari DES dapat dilihat pada Gambar 3.

Algoritma enkripsi dengan DES dapat dilihat pada Gambar 4. Pada Gambar 4 terlihat bahwa permutasi atau perulangan yang digunakan cukup kompleks sehingga disetujui oleh National Bureau of Standard (NBS) setelah penilaian kekuatannya oleh National Security Agency (NSA) Amerika Serikat.

Implementasi algoritma DES pada penggunaan *password* di Linux adalah *password* yang digunakan oleh *user* dijadikan input untuk dipermutasi sesuai skema DES pada Gambar 4 untuk mendapatkan cipherteks yang aman, kemudian digunakan *salt* untuk mendapatkan hasil yang optimal. *Salt* adalah string dengan dua buah karakter yang merupakan himpunan dari [a-zA-Z0-9./]. String ini digunakan untuk mendekrip *password* yang tersimpan dalam 4096 cara yang berbeda.



Gambar 3. Skema Global Algoritma DES

Dewasa ini banyak program yang digunakan untuk mendapatkan *password user*. Salah satu program terbaik yang tersedia untuk mengcrack *password* UNIX adalah John The Ripper [8].

4. JOHN THE RIPPER

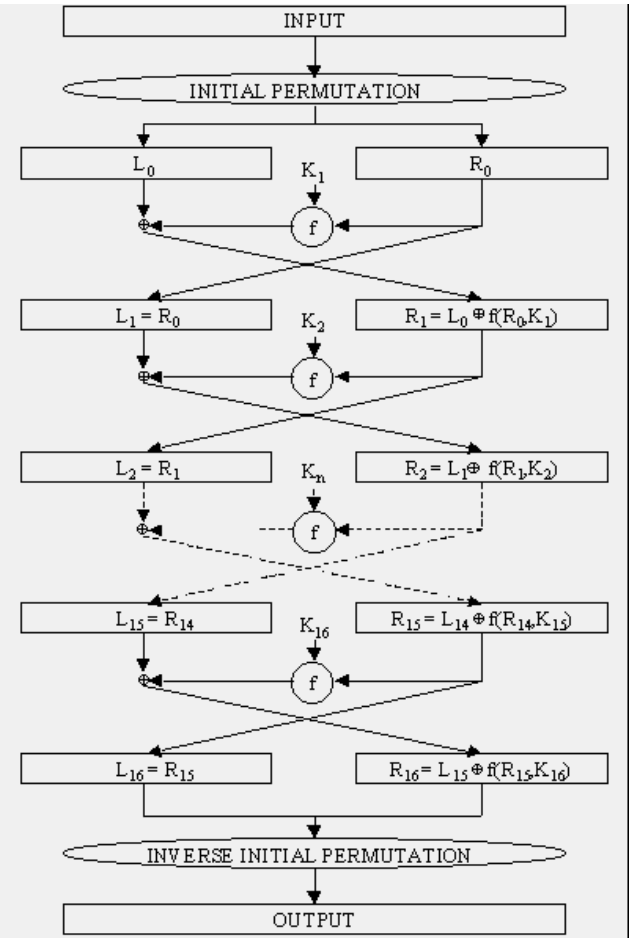
4.1 Apa Itu John The Ripper

John The Ripper adalah suatu *cracker password* yang cepat yang tersedia pada banyak platfom, antara lain UNIX, Windows, DOS, BeOS dan OpenVMS. Tujuan utama dari John The Ripper adalah untuk mendeteksi kelemahan *password* pada sistem UNIX (termasuk Linux) [8]. John The Ripper merupakan program yang dapat membantu administrator menentukan kelayakan suatu *password*. Namun John The Ripper juga digunakan oleh pihak-pihak yang tidak berkepentigan untuk mendapatkan *password* seorang *user*. Situs resmi John The Ripper berada di www.openwall.com/john.

John The Ripper adalah tools untuk mengcrack yang opensource paling popular di dunia dan telah dirancang khusus untuk mengcrack sebanyak mungkin *password* dalam waktu yang sangat singkat [3][4].

John The Ripper mampu menangani berbagai type enkripsi yang dikenakan terhadap *password*, dan John The Ripper juga menyediakan fasilitas untuk membuat

permutasi dari setiap kata yang ada di dalam *wordlist* atau kamusnya. Teknik enkripsi yang didukung oleh John antara lain : DES, MD5, dan lain-lain.



Gambar 4. Skema Global Algoritma DES

John The Ripper juga tersedia dalam bentuk komersial yaitu John The Ripper Pro yang dapat diperoleh di http://www.openwall.com/john/pro/

Wordlist dari program John The Ripper juga dapat ditambahkan, disamping yang telah disediakann oleh pihak pengembang John The Ripper sendiri di www.openwall.com/wordlists.

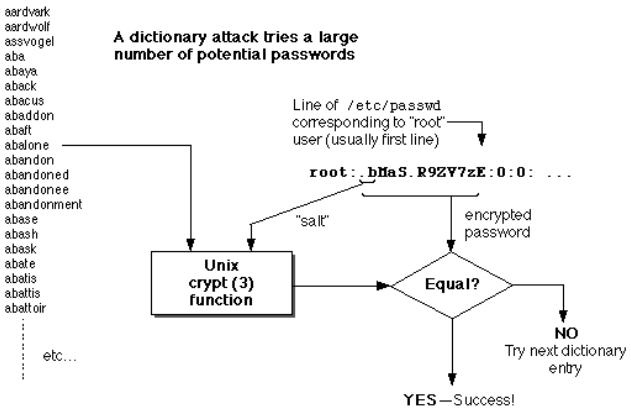
Secara umum, John The Ripper mendukung dan dapat mendeteksi tipe crypt dari sistem UNIX berikut :

1. traditional and double-length berbasis DES
2. SDI extended berbasis DES
3. FreeBSD berbasis MD5 (yang sekarang juga digunakan di dalam Linux dan Cisco IOS)
4. OpenBSD berbasis Blowfish (sekrang juga digunakan dalam beberapa distribusi Linux)
5. John The Ripper juga mendukung Kerberos/AFS dan hash Windows LM berbasis DES[4]

4.2 Cara Kerja John The Ripper

John The Ripper bekerja dengan cara membaca file teks yang berisi *password* yang telah dienkripsi sistem dan di bandingkan dengan enkripsi dari kata-kata di dalam kamus yang ada pada John The Ripper (biasanya John The Ripper menyediakan kamus yang sering digunakan sebagai *password* yang mudah didapat secara *online*) dan setiap enkripsi yang dilakukan terhadap kata-kata di dalam kamus dicocokan dengan enkripsi yang ada pada *password user*.

Jika ada enkripsi yang sesuai, maka *password* dari *user* tersebut adalah kata di dalam kamus itu yang enkripsinya sesuai dengan enkripsi yang ada pada kamus John The Ripper. Hal ini diilustrasikan pada Gambar 5.



Gambar 5. Prinsip kerja John The Ripper

John The Ripper bekerja dalam mode-mode sebagai berikut :

a. Wordlist : John The Ripper menggunakan sebuah file yang teridir atas list kata yang akan melakukan pemeriksaan terhadap *password*.
b. Single crack : pada mode ini, John The Ripper akan mencoba untuk mengcrack *password* menggunakan informasi login/GECOS sebagai *password* untuk masuk
c. Incremental : mode ini adalah merupakan mode yang sangat powerful, tangguh, John The Ripper akan mencoba kombinasi karakter apapun untuk mendapatkan *password*

Ketika program john dijalankan maka, program ini akan membuat sebuah directory di ~/.john tempat meletakkan hasil dekrispsinya. Biasanya dengan ekstensi .pot.

Maka untuk dapat melihat isi file.pot ini, dilakukan perintah

```
root@router#john –show file.pot
```

Jika john the ripper dapat meretas *password* sistem, maka kita akan mendapatkan jawaban dengan *user*name diikuti *password* dari *user* tersebut, misalnya :

```
root: password1
hary: password22
```

4.3 John The Ripper dan Keamanan Linux

Sitem LINUX yang baru telah mengembangkan sistem keamanan yang lebih baik. Ini terbukti dengan menggunakan algoritma yang sama sekali berbeda yaitu SHA, hingga saat ini, telah digunakan SHA2 untuk enkripsi *password user* di Linux. Hal lain mengenai *password* pada sistem Linux adalah apabila seseorang dapat mengakses sebuah mesin Linux (terlebih secara hardware), maka ia dapat melakukan apa saja terhadap mesin Linux tersebut, karena jika grub-nya dibiarkan secara default, maka sesorang dapat masuk ke grub tersebut. Ketika mesin dijalankan seseorang dapat merubah konfigurasi grub untuk menjalankan single *user*.

Jadi walaupun sesorang tidak mengetahui *password* root dari sistem, tetapi ia dapat mereset *password* root tersebut setelah memasuki system yang masuk ke mode single *user*.

Untuk sistem *password* sendiri, di Linux terdapat cara untuk mengetahui algoritma apa yang digunakan oleh suatu sistem Linux tersebut dalam mengenkripsi *password*nya. Hal ini dapat di lihat di `/etc/shadow` (untuk sistem Linux terbaru) yang hanya dapat dijalankan oleh root :

```
root@router#cat /etc/shadow
root:$1$xjp8B1D4$tyQNzvYCIrf1M5RYhAZlD.:1
4076:0:99999:7:::
daemon:*:14063:0:99999:7:::
***hasil /etc/shadow***
```

Jika setelah *user*name terdapat tanda :

1. 1 maka algoritma yang digunakan adalah MD5
2. $2a$ maka algoritma yang digunakan adalah blowfish
3. 5 dan 6 maka algoritma yang digunakan adalah SHA

John The Ripper sendiri tidak mendukung dalam hal mendekripsi SHA dan variannya. Diperlukan program antara atau juga *patch* agar John The Ripper dapat mendekripsi *password* tersebut. Dalam hal ini Penulis tidak melakukan dekripsi terhadap algoritma SHA ini, karena batasan masalah pada makalah ini adalah algoritma yang digunakan adalah DES.

Algoritma DES dalam implementasinya di Linux memiliki kekurangan yaitu menggunakan program crypt yang dianggap lemah.

4.4 Bagaimana John The Ripper Meretas *Password User* Linux

Pertama-tama lihat file `/etc/passwd`. Pada sistem Linux generasi awal, *password* di simpan di file /etc/passwd yang dapat di lihat oleh siapa saja, namun tetap dienkripsi. Namun hal ini tentu tidak cukup. Walaupun yang berhak melakukan perubahan terhadap file /etc/passwd hanya root dan *user* lain hanya bisa melihat file tersebut, namun dewasa ini, bagian *password* di simpan di dalam file /etc/shadow atau /etc/master.shadow yang memilki akses hanya seorang root, dan tidak seorang *user* biasapun dapat melihat file /etc/shadow tersebut. Karena alasan inilah kita perlu akses root untuk melihat file /etc/shadow

Secara umum, struktur atau format dari file /etc/passwd adalah sebagai berikut [5]:

`account:`*`password`*`:UID:GID:GECOS:directory:shell`

Untuk sistem UNIX yang lama, karena tidak menggunakan /etc/shadow, maka tulisan `:`*`password`*`:` hasil enkripsi dari *password* diletakkan pada file /etc/passwd.

Pada versi Linux yang baru, yang mendukung file shadow, kata `:`*`password`*`:` digantikan dengan `:x:`. Dan penyimpanan *password* diletakkan di /etc/shadow. Sedangkan untuk format /etc/shadow sendiri adalah sebagai berikut [6]:

`user``name:passwd:last:may:must:warn:expire:disable:reserved`

Pada file /etc/shadow ini, kata `:`*`password`*`:` mengandung beberapa bagian yang dipisahkan oleh tanda $. Tanda $ ini mengisyaratkan bahwa Linux ini mendukung *Modular Crypt Format* (MCF). MCF menentukan skema format *password* yang dapat digunakan oleh berbagai algoritma. Dewasa ini MCF adalah format paling populer untuk melakukan enkripsi di sistem UNIX. Berikut adalah tabel yang menjelaskan 3 bagian yang sesuai dengan format MCF :

Tabel 1 Penjelasan format yang ada dalam /etc/shadow

Bagian	Fungsi	Penjelasan
1	Algoritma	Jika 1 maka MD5 Jika 2 maka Blowfish
2	Salt	Nilai random sebagai input untuk menciptakan *password* yang unik walaupun *password*nya sama
3	Encrypted *Password*	Hash dari *password user*

Kekuatan suatu *password* dapat ditentukan dengan seberapa besar waktu yang diperlukan dan tenaga yang diperlukan untuk melakukan dekripsi terhadap *password* tersebut.

5. CRYPT

Pada saat *user* memanggil perintah `/bin/passwd`, biasanya untuk mengganti *password* atau mereset

password, maka perintah tersebut akan memanggil fungsi lain yaitu Crypt yang terdapat di dalam `/bin/crypt`.

Crypt adalah fungsi untuk melakukan enkripsi terhadap *password* [7]. Crypt berdasarkan pada algortima *Data Encryption Standard* (DES) dengan berbagai variasi yang bertujuan untuk mendukung penggunaan enkripsi terhadap implementasi hardware. Crypt menggunakan salt. Salt seperti yang sudah dijelaskan didepan, terdiri atas dua buah karakter yang merupakan himpunan dari [a-zA-Z0-9./] yang dapat menghasilkan penyimpanan dalam 4096 cara yang berbeda

Dengan mengambil 7 bit dari setiap karakter dari setiap masukan *password*, didapatkan 56-bit key. Key yang berjumlah 56 bit ini digunakan untuk mengenkripsi secara berulang sebuah string yang konstant, biasanya string yang mengaundung semuanya angka nol. Kembalian dari *password* yang terenkripsi ini adalah kumpulan dari 13 karakter ASCII yang dapat di print, yang mana dua buah karakter didepan menggambarkan saltnya sendiri.

Perlu diperhatikan bahwa ruang solusi dengan penggunaan crypt mencapai 7.2e16 kemungkinan nilai.[10] Penggunan *exhaustive searches* terhadap kunci ini mungkin dilakukan jika menggunakan teknologi komputer massa yang paralel. Hal ini tentu merupakan suatu hal yang tidak diharapkan, karena *password* dapat dipecahkan walaupun harus menggunakan teknologi tersebut.

Dewasa ini peningkatan keamanan terhadap algoritma autentifikasi *user* Linux telah dikembangkan sehingga diharapkan kemanan sistem passsword akan sulit untuk dibobol[2].

6. STUDI KASUS

Pada bagian ini akan dilakukan pengujian terhadap keamanan sistem *password* yang ada pada beberapa sistem Linux.

6.1 Perangkat Uji

Percobaan ini penulis lakukan di sebuah komputer laboratorium dasar yang memiliki Linux fedora. Selain itu digunakan juga sebuah komputer dengan Linux didalamnya namun dengan versi yang lebih lama.

Pertama kali harus dilihat dulu, apakah program yang diperlukan sudah diinstall.

```
root@router#john
no file or program
```

Jika belum lakukan perintah berikut (khusus untuk Linux fedora, telah disediakan yum, yaitu sebuah paket installer yang menangani masalah penginstalan seperti *dependency* dan lain sebagainya).

```
root@router#yum install john
```

Jika kita telah mendapatkan program John The Ripper, maka ketika kita mengetikan

```
root@router#john
```

Jika program telah berhasil di install, maka akan keluar cara penggunaan program John The Ripper ini.

Studi kasus akan memakan waktu yang lama dan sistem akan mendapatkan beban yang berat karena program ini akan banyak memakan *resource* sistem sehingga ada kemungkian sistem tidak merespon, terutama jika terdapat banyak program yang dijalankan secara background.

Skema pengujian terbagi dua, yaitu skema 1 dengan sistem Linux baru, dan skema 2 dengan sistem Linux yang lama.

6.2 Pengujian dengan Skema 1

Pada skema 1 ini, sistem Linux yang digunakan adalah Linux dengan kernel 2.6.x, dengan kata lain Linux versi terbaru. Sebelum melangkah lebih lanjut, lihat dahulu, sistem Linux yang digunakan dengan mengetikan perintah

```
root@router#uname –a
```

Didapatkan hasilnya bahwa Linuxnya adalah fedora 11.

Selanjutnya kita akan gunakan program shadow dari john the ripper untuk mengabungkan file /etc/passwd dan /etc/shadow yang diletakan terpisah.

```
root@router#cd
root@router#/usr/bin/unshadow
/etc/passwd /etc/shadow > password.db
```

Maka program ini akan buat sebuah file pada directory $JOHN/john.pot.

Setelah itu, dilakukan perintah dibawah ini untuk mendekripsi file *password*.db yang berisi *password* dari *user*.

```
root@router#john password.db
No password hashes loaded
```

Ternyata setelah penulis melihat ke /etc/shadow

```
root@router#cat /etc/shadow
```

Hasil yang diperoleh adalah, terdapat tandan 6 yang menyatakan bahwa Linux Fedora 11 menggunakan algoritma SHA yang tidak dapat didekripsi langsug oleh john The ripper.

Sejauh ini dapat disimpulan bahwa Linux versi baru dengan algoritma enkripsi *password*nya lebih baik sulit untuk didekripsi. Untuk melihat algoritma yang digunakan dapat melihat ke file /etc/shadow.

6.3 Pengujian dengan skema 2

Sama seperti langkah pengujian dengan skema 1, pada skema 2, hal yang harus dilakukan adalah :

```
root@router#cd
```

```
root@router#/usr/bin/unshadow
/etc/passwd /etc/shadow > password.db
```

Setelah mendapatkan file *password*.db selanjutnya, penulis mencoba menjalankan John The Ripper dalam mode single:

```
root@router#john --single passsword.db
```

Kemudian penulis mencoba menjalankan John The Ripper dalam mode wordlist:

```
root@router#john -wordfile:wordlistku.txt password.db
```

Jika tidak digunakan opsi apapun maka john the ripper akan mencoba ketiga mode yang ada yaitu, pertama kali mencoba single mode, kemudian wordlist mode lalu baru menggunakan incremental mode.

```
root@router#john password.db
```

Output dari eksekusi ini adalah:

```
john /tmp/crack.password.db
Loaded 1 password (FreeBSD MD5 [32/32])
```

Proses ini memakan waktu yang lama. Setelah selesai, untuk melihat hasil *password* yang didekripsi dapat digunakan perintah:

```
root@router#john –show password.db
test:123456:1002:1002:test,,,:/home/test:/b
in/bash
hary:abc123:1003:1003::/home/hary:/usr/bin/
rssh
2 passwords cracked, 1 left
```

Output diatas menunjukan dengan jelas bahwa *user* `test` mempunyai *password* 123456 dan *password* `hary` memiliki *password* abc123.

6.4 Hasil Pengujian

Setelah dilakukan pengujian, didapatkan hasil bahwa program John The Ripper dapat bekerja dalam sistem Linux yang tidak terlalu baru. Sedangkan Linux dengan keluaran terbaru menggunakan algoritma yang berbeda yang tidak dapat langsung dipecahkan oleh johnthe ripper, tetapi tetap dapat dilakukan jika menggunakan program tambahan atau *patch* yang menconvert hasil dari /etc/passwd menjadi format yang dapat dikelola oleh John The Ripper.

7. KESIMPULAN

Keamanan *password* menjadi sangat penting di era yang semakin cangih ini. Untuk mencegah penyalah gunaan oleh pihak yang tidak diinginkan. Pengujian terhadap kemanan *password user* lokal Linux dengan menggunakan program John The Ripper menunjukan bahwa *password user* dapat diketahui jika teknik enkripsi yang digunakan belum diupdate karena masih menggunakan Linux versi lama dan /etc/shadownya dapat diakses dan digunakan teknik *bruteforce* untuk mendapatkan *password* seorang *user*.

File /etc/passwd harus tetap dijaga agar dapat dibaca oleh siapapun, karena banyak program dari sistem Linux yang menggunakan /etc/passwd. misalnya jika file /etc/passwd di seting agar tidak seorangpun dapat melihatnya, maka ketika perintah ls -l dijalankan, tidak terdapat nama dari pemilik suatu file, namun yang ada adalah nomor ID dari *user* tersebut. Yang harus ditingkatkan adalah *password*. Program `passwd` akan melakukan pengecekan terhadap *password user* yang mudah dicrack sehingga program ini dapat menyarankan untuk menggunakan *password* yang lebih kuat.

Saat ini semakin banyak variasi algoritma yang digunakan untuk menyimpan data *password* dari seorang *user*. Namun demikin kita juga tetap harus waspada, jangan sampai *password* kita diberikan kepada orang lain, jangan pula kita menggunakan *password* yang mudah ditebak.

REFERENSI

[1]http://www.Linux.org/docs/ldp/howto/Security-HOWTO/*password*-security.html

[2] Stallings, William *Cryptography and Network Security Principles and Practices, Fourth Edition.* Prentice Hall 2005

[3] Seifried, Kurt *Linux Administrators Security Guide* https://www.seifried.org/lasg 1999

[4] Herzog, Pete *Hacking Exposed Linux: Linux Security Secrets & Solutions*. McGraw-Hill 2008

[5]http://www.cyberciti.biz/faq/understanding-etcpasswd-file-format

[6] http://www.cyberciti.biz/faq/understanding-etcshadow-file

[7]http://www.kernel.org/doc/man-pages/online/pages/man3/crypt.3.html

[8]http://www.nic.com/~dave/SecurityAdminGuide/SecurityAdminGuide.html#toc9

[9] Munir, Rinaldi, “Kriptografi”, Informatika ITB.

[10] http://tldp.org/HOWTO/Shadow-*Password*-HOWTO-2.html