Laporan Resmi Praktikum Keamanan Data

“Telnet dan SSH Server”

Dimas Rizky H.P. – 2110141011 – 3 D4 IT A

Percobaan : Cara kerja telnet & SSH Server, SSH no password, cara kerja SSH bruteforce attack dengan medusa, penangkal bruteforce menggunakan program fail2ban, perbandingan dan analisa anomali SSH bruteforce attack.

Laporan Pendahuluan :

1. Cara Kerja Telnet

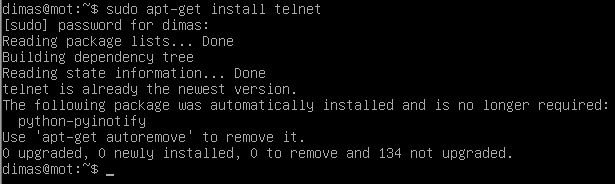
***Dasar Teori :***

Telnet adalah singkatan dari *Telecommunications network protocol.* Telnet adalah salah satu aplikasi internet paling tua yang pernah ada. Telnet memungkinkan kita untuk menghubungkan “terminal” kita dengan *host remote* yang berada di luar jaringan. Pada masa ARPANET sebelum adanya workstation grafis, atau personal komputer, setiap orang menggunakan *terminal* yang terhubung dengan *mainframe* atau *minicomputer* melalui koneksi serial. Setiap terminal memiliki *keyboard* sebaga masukan dan monitor sebagai media untuk mencetak keluarannya.

Telnet biasa digunakan untuk *remote login* dari PC ke PC lain dalam satu jaringan. *Remote login* Telnethanya menyediakan koneksi *text only,* yang biasanya ditampilkan dalam bentuk *command line prompt.* Telnet merupakan aplikasi *client/server*. *Client* mengambil karakter yang dimasukkan dari *keyboard,* mengirimkannya ke *server* dan mencetak *output* yang dikirim oleh *server.* Sedangkan *server* bertugas untuk melewatkan input dari client ke sistem.

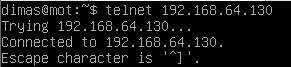
***Langkah percobaan :***

1. Install package telnet dengan perintah *$sudo apt-get install telnet*



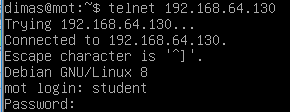
*Gambar 1.1 instalasi package telnet*

1. Setelah menginstall package, untuk melakukan koneksi telnet ke target dapat dilakukan dengan menggunakan perintah $telnet <IP address server>



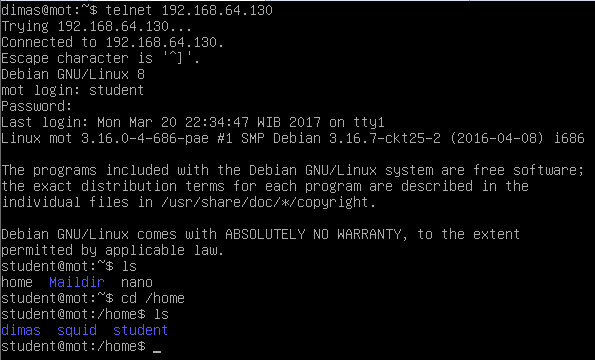
*Gamar 1.2 percobaan koneksi telnet*

1. Jika koneksi berhasil, akan diminta username dan password untuk login ke mesin server



*Gambar 1.3 autentikasi untuk login ke komputer server*

1. Telnet berhasil dilakukan dan server dapat diakses melalui terminal



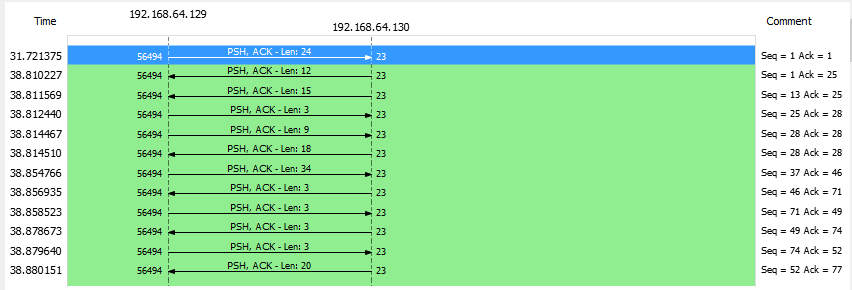
*Gambar 1.3 telnet berhasil, sehingga client dapat masuk ke komputer server*

***Analisa :***

Seperti yang sudah dibahas pada laporan pendahuluan, telnet dikirimkan menggunakan plain-text per karakter yang diketik pada client, ke servernya. Karakter tersebut tidak dienkripsi, melainkan dikirimkan melalui internet dalam format NVT character set.

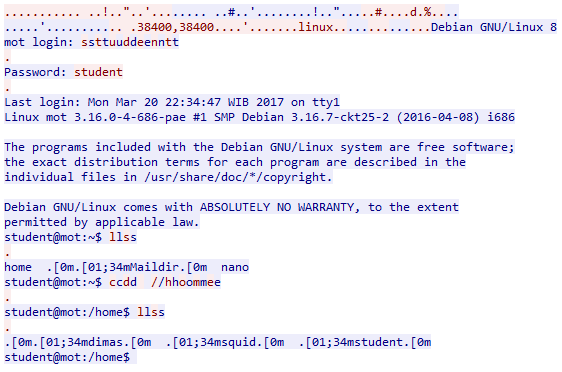
Pada saat packet sampai ke server (192.168.64.130), NVT tadi di konversi menjadi character set milik remote computer. Oleh server di proses perintah yang dikrimkan oleh client, lalu responnya kembali dikirimkan ke client melalui proses yang sama menggunakan character NVT melalui internet. Begitu seterusnya sampai dengan kata kata ‘exit’ atau identifier lain yang dapat menghentikan koneksi telnet dikirimkan.

Pada packet yang tertangkap pada wireshark, dapat dilihat diagram sebagai berikut :



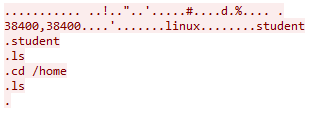
*Gambar 1.4 TCP flow graph telnet*

Pada diagram TCP di atas, dapat dilihat, adanya pertukaran data yang terjadi antara IP Client(192.168.64.129) dan server(192.168.64.130). Data stream SYN-ACK sangat banyak dilakukan, hal itu disebabkan karena untuk setiap character yang diketikan, packet berisi character itu akan segera dikirimkan saat itu juga. Sehingga dalam satu sesi telnet, akan ada sangat banyak packet yang dikirimkan dan diterima baik oleh client maupun server telnet. Untuk detail setiap, TCP stream, dapat diliat pada gambar di bawah ini



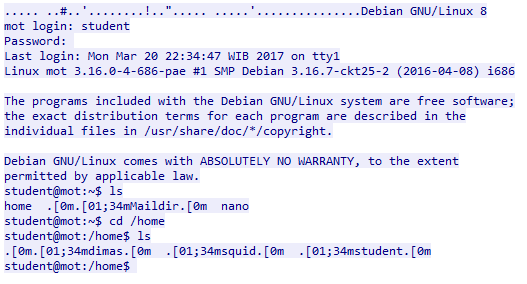
*Gambar 1.5 TCP Stream*

Lebih jelasnya lagi, untuk pertukaran data dari IP address .129 🡪 .130 adalah sebagai berikut



*Gambar 1.6 TCP Stream client🡪server*

Pada gambar 1.6, ditunjukan apa saja yang dikirimkan oleh client ke server, dan bisa dilihat secara jelas, konten apa yang dikirimkan. Mulai dari autentikasi dan perintah perintah yang dieksekusi. Sedangkan pada TCP stream server🡪client, packetnya adalah sebagai berikut :



*Gambar 1.7 TCP Stream server🡪client*

Data yang dikirimkan dari server ke client juga dapat dilihat secara mudah lewat packet yang dikirimkan melalui TCP. Menandakan, jika ada yang menyadap atau melakukan *intercept* paket telnet ini, password, username dan informasi informasi lainnya dapat diketahui dan dicuri secara mudah. Yang berarti keamanan dari protocol telnet ini masih sangat kurang.

1. Cara Kerja SSH

***Dasar Teori :***

SSH adalah suatu protokol yang memfasilitasi sistem komunikasi yang amana antara dua sistem yang menggunakan arsitektur *client/server*, serta memungkinkan seorang user untuk login ke server secara remote.

SSH dirancang untuk menggantikan *service-service* di sistem *unix.linux* yang menggunakan sistem *plain-text*. Bedanya, *plain-text* yang digunakan akan dienkripsi terlebih dahulu. Artinya apabila percakapan/komunikasi tersebut disadap, meskipun informasi diketahui orang, namun bentuk informasi tersebut masih berupa data yang terenkripsi.

SSH merupakan pengembangan dari telnet dengan memperkuat sisi keamanannya saja. Protokol SSH menyediakan layanan :

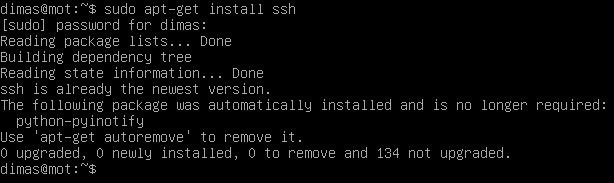
* Pada saat awal terjadi kooneksi, client melakukan pengecekan apkah host yang dihubungi sudah terdaftar pada client atau belum
* Client mengirimkan proses autentifikasi ke server menggunakan teknik enkripsi 128 bit
* Semua data yang dikirimkan dan diterima menggunakan teknik enkripsi 128 bit sehingga sangat sulit dibaca tanpa mengetahui kode enkripsinya.

Metode enkripsi SSH adalah dengan cara menyimpan key-pair antara client dan server di dalam direktori .ssh/ pada file *known\_host.* Hal ini memungkinkan untuk secara manual melakukan copy key secara langsung dari key yang telah tergenerate dari client ke file *.ssh/known\_host* yang berada di server tujuan.

Berkat hal tersebut, dimungkinkan adanya koneksi SSH antara client dan server tanpa diperlukan adanya proses autentikasi lagi setiap kali membuka koneksi SSH.

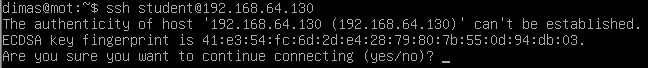
***Langkah percobaan 1 – Cara Kerja SSH :***

1. Install package SSH dengan menggunakan perintah $sudo apt-get install ssh



*Gambar 2.1 Instalasi package SSH*

1. Setelah itu, untuk melakukan koneksi ssh ke server dapat menggunakan perintah $ssh <namauser>@<alamat ip komputer server>
2. Setelah mengeksekusi perintah tersebut, akan diminta konfirmasi tentang penulisan key public yang akan dilakukan pada client.



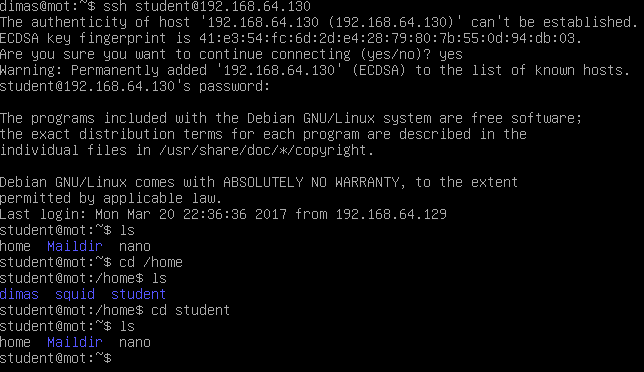
*Gambar 2.2 konfirmasi pembuatan public-key di sisi client (file .ssh/known\_hosts)*

1. Setelah penulisan public key selesai, maka selanjutnya adalah proses autentikasi user dan password untuk login ke server



*Gambar 2.3 Autentikasi diperlukan untuk login SSH*

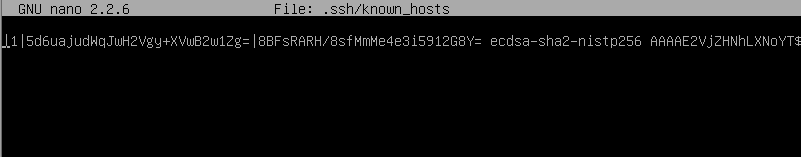
1. Setelah itu, server akan dapat diakses oleh client melalui terminal.



*Gambar 2.4 SSH berhasil dilakukan, client masuk ke terminal server.*

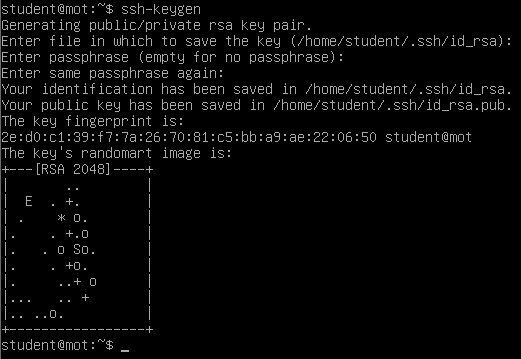
***Langkah percobaan 2 – SSH no Password :***

1. Hapus terlebih dahulu konten pada file *.ssh/known\_host* pada server jika sudah pernah melakukan koneksi SSH ke server tersebut.



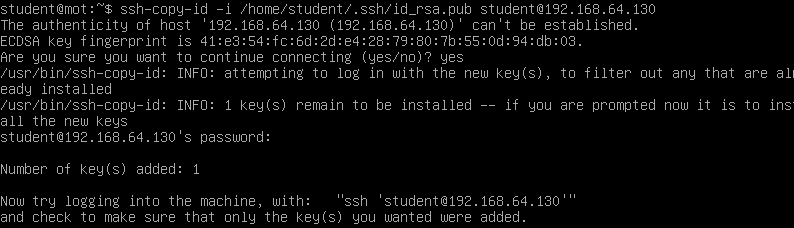
*Gambar 2.5 Hapus key fingerprint yang sudah ada dengan server yang ada*

1. Buat SSH Keygen dengan perintah $ssh-keygen



*Gambar 2.6 Generate SSH keygen*

1. Setelah SSH keygen tergenerate, copy kan public-key yang tergenerate tersebut ke computer server dengan perintah $ssh-copy-id –i /home/<namauser>/.ssh/id\_rsa.pub <namauser>@<ip server>



*Gambar 2.7 Proses copy file rsa\_key hasil generate ke SSH server*

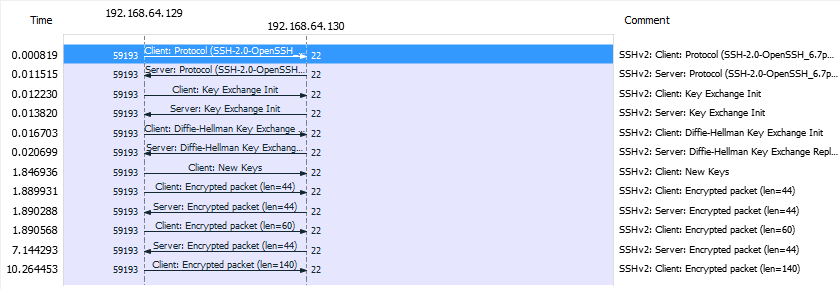
1. Jika setting berhasil, pada saat melakukan koneksi SSH ke server bersangkutan, proses autentikasi akan dilompati.



*Gambar 2.8 Koneksi SSH tanpa password berhasil dilakukan*

***Analisa :***

Pada percobaan pertama, percobaan SSH, packet yang ditangkap pada saat SSH dijalankan adalah sebagai berikut (dalam TCP flow diagram) :



**53**

**43**

**3**

**2**

**1**

*Gambar 2.9 TCP flow diagram*

Pada gambar 2.9, label nomor dua ditunjukan ada dua buah packet, yaitu client key exchange init, dan server key exchange init. Ini merupakan fase pada saat penukaran key fingerprint, setelah konfirmasi exchange key disetujui oleh client. Setelah itu pada label nomor tiga, ada packet yang mengkonfirmasi, algoritma apa yang akan digunakan untuk melakukan enkripsinya nanti dan juga ada pengiriman public key server ke client.

Setelah algoritma dikonfirmasi dan public key server diterima client, beranjak ke label nomor 4, terdapat pengiriman public-key client yang sudah dienkripsi ke Server untuk memulai proses enkripsinya. Pada label nomor lima, itu merupakan data yang sudah dienkripsi menggunakan algoritma yang sudah disepakati pada label nomor 3 tadi.

Detail dari TCP diagram dapat ditemui pada TCP stream di bawah ini :

*Gambar 2.10 TCP Stream packet SSH*

Pada TCP stream dapat dilihat, pada gambar sebelah kanan, dua baris di awal adalah proses konfirmasi versi SSH yang digunakan, lalu baris setelahnya adalah konfirmasi pertukaran key dan algoritma enkripsi yang digunakan. Pada gambar sebelah kanan, dapat dilihat packetnya tidak beraturan dan terlihat acak. Inilah hasil enkripsi karakter yang dilakukan oleh logika enkripsi yang sudah disetujui tadi

Dengan ini cukup membuktikan, bahwa meskipun packet dapat di *intercept*, konten yang ada di dalam packet tidak akan diketahui maksudnya oleh penyada packet tersebut. Menandakan bahwa sistem keamanan protocol SSH sudah cukup kuat terhadap penyadap packet.

Pada proses login tanpa menggunakan password, request pertukaran key tidak ada yang berbeda, namun bedanya, untuk proses autentikasi, sudah diganti menggunakan rsa\_key sehingga tidak diperlukan lagi proses autentikasi.

1. Cara Kerja SSH bruteforce attack dengan medusa

***Dasar teori :***

Bruteforce attack adalah metode untuk meretas password dengan cara mencoba kemungkinan kombinasi yang ada pada “wordlist”. Metode ini dijamin akan berhasil menemukan password yang ingin diretas, selama waktu yang diperlukan cukup dan tidak ada program yang menghalangi proses peretasan.

Medusa merupakan salah satu contoh aplikasi linux yang dapat digunakan untuk melanacarkan SSH bruteforce attack dengan cara menyuplai username dan password dalam *wordlist*.

***Langkah percobaan :***

1. Lakukan instalasi medusa dengan perintah $sudo apt-get install medusa



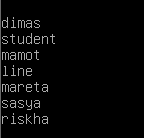
*Gambar 3.1 instalasi package medusa*

1. Setelah proses instalasi berhasil, siapkan dua buah file wordlist, wordlistuser.txt yang berisi list username yang ingin di test, lalu wordlistpass.txt yang berisi list password yang ingin dikombinasikan.



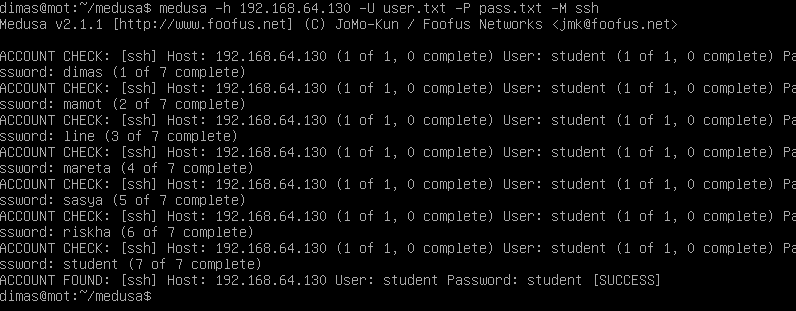
*Gambar 3.2 Siapkan wordlist untuk username dan password*

1. Pada masing masing file wordlist, masing-masing kata dipisahkan dengan baris baru.



*Gambar 3.3 Contoh penulisan list password/user pada wordllist*

1. Setelah itu, untuk melakukan bruteforce attack dapat dijalankan dengan perintah *$medusa –h <ip\_address\_target> -U <file wordlistuser.txt> -P <file wordlistpass.txt> -M ssh*

**

*Gambar 3.4 proses bruteforce attack oleh medusa*

1. Medusa akan melakukan proses bruteforce attack dengan satu per satu kombinasi password dan username yang ada di masing-masing wordlist.

***Analisa :***

Medusa akan coba mencoba melakukan koneksi SSH ke server SSH menggunakan semua kombinasi yang memungkinkan antara wordlist user dan password. Pada percobaan ini, list user yang ada pada wordlist adalah satu, sedangkan list password ada tujuh password. Sehingga, medusa akan mencoba melakukan penetrasi sebanyak tujuh kali penebakan password.

Namun, pada packet yang ada pada wireshark, koneksi SSH yang dibangkitkan hanya ada 3 sesi SSH. Dapat dillihat di bawah ini :



*Gambar 3.5 sesi SSH yang terbuka*

Hal ini terjadi, karena setiap sesi SSH, SSH menyediakan tiga kali kesempatan menebak password, sehingga untuk tujuh kali penebakan, sesi SSH yang dibuka hanya cukup tiga kali koneksi SSH. Dalam segi kecepatan, medusa melakukan tujuh kali penetrasi password hanya dalam waktu 29 detik.

1. Menangkal brutefoce attack menggunakan program fail2ban

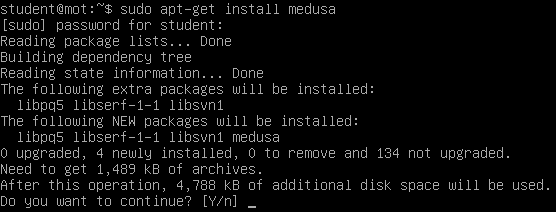
***Dasar Teori :***

Bruteforce attack dapat ditangkal dengan program bernama fail2ban. Fail2ban akan mendeteksi adanya anomali proses login SSH yang dapat dihasilkan karena penggunaan aplikasi untuk bruteforce seperti medusa atau hydra.

Fail2ban akan bekerja dengan cara melakukan block terhadap alamat IP tertentu dalam beberapa waktu yang ditentukan dalam file konfigurasi yang ada.

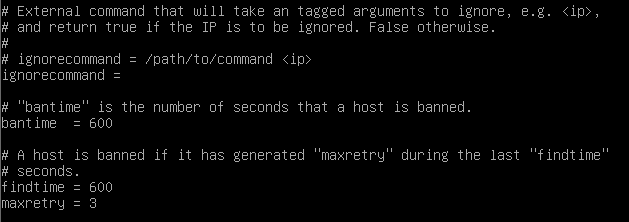
***Langkah percobaan :***

1. Lakukan instalasi package fail2ban menggunakan perintah *$sudo apt-get install fail2ban*

**

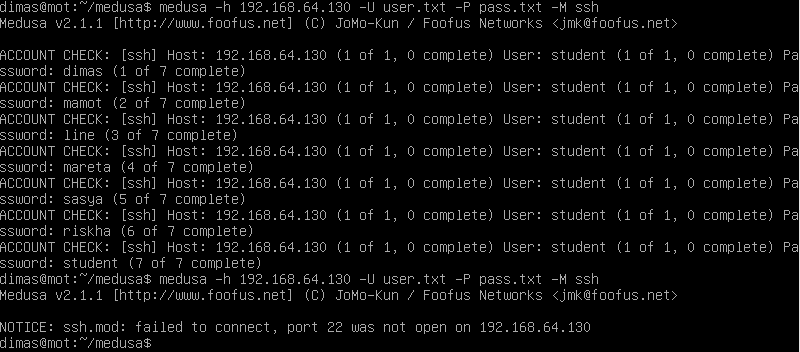
*Gambar 4.1 instalasi package fail2ban*

1. Buka file konfigurasi fail2ban untuk melihat konfigurasi yang sedang aktif, secara default, fail2ban akan melakukan pemblokiran pada IP address yang melakukan koneksi sesi SSH sebanyak 3kali dalam rentang waktu 5 menit selama 600 second

**

*Gambar 4.2 konfigurasi fail2ban yang ada pada /etc/fail2ban/jail.conf*

1. Dapat dilihat pada sisi penyerang, setelah 5 kali tries bruteforce, port 22 SSH server akan otomatis ditutup oleh server.

**

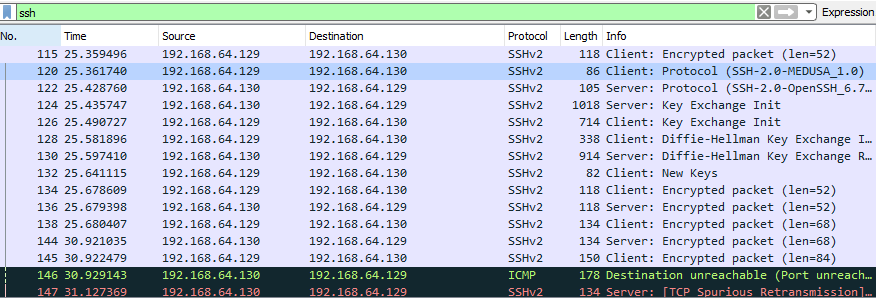
*Gambar 4.3 Penutupan/pemblokiran port SSH pada server SSH oleh fail2ban*

***Analisa :***

Mulai pada gambar 4.2, kita dapat mengetahui konfigurasi fail2ban yang aktif. Pada fail2ban, akan melakukan banning pada IP address yang melakukan koneksi SSH yang sekurang kurangnya melakukan tiga sesi SSH dalam rentang waktu 5 menit. Banning akan terjadi selama lima menit.

Pada gambar 4.3, kita juga dapat melihat dilakukan bruteforce dengan kombinasi tujuh password. Seperti yang kita tahu pada percobaan bruteforce sebelumnya, untuk mencoba tujuh kombinasi password sesi SSH yang dibutuhkan adalah 3 kali sesi, dan pada percobaan tiga sebelumnya, kita mengetahui bahwa medusa membutuhkan waktu hanya 29 menit untuk mencoba tujuh kombinasi password.

Kondisi ini memenuhi kondisi yang ada pada file konfigurasi program fail2ban. Sehingga fail2ban akan melakukan banning pada IP komputer penyerang. Pada wireshark, dapat dilihat, setelah dibukanya gerbang SSH sesi ketiga, port 22 SSH pada SSH server ditutup oleh program fail2ban, berikut gambarnya :



*Gambar 4.4 algoritma fail2ban melakukan penutupan port 22 untuk IP 192.168.64.129*

*Sehingga, pada saat ingin melakukan koneksi SSH kembali ke SSH server, munculah prompt seperti gambar 4.3, yaitu gagal, karena port SSH 22 di tutup untuk IP penyerang.*

1. Perbandingan dan analisa anomali packet pada saat dilakukan bruteforce attack

***Dasar Teori :***

Seerangan SSH bruteforce pada dasarnya adalah rangkaian percobaan login SSH yang sama metodenya dengan login SSH secara normal. Packet yang dikirimkan dan diterima juga bergerak dalam suatu pola tertentu. Yang membedakan antara normal SSH login dengan bruteforce attack adalah waktu antara jeda pengiriman packetnya jika password yang digunakan salah.

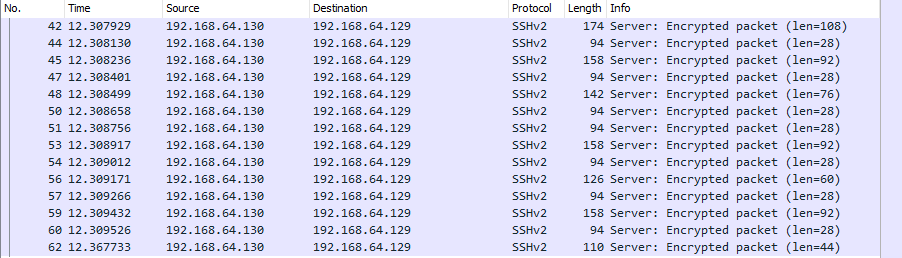
Perbedaan waktu jeda antara masing-masing percobaan login ini dinamakan anomali. Normalnya normal SSH login membutuhkan waktu beberapa detik agar mendapatkan akses untuk melakukan SSH. Namun dengan bruteforce attack, waktu jeda antar percobaan login akan jauh lebih pendek dibandingkan dengan SSH login secara normal. Percobaan ini akan membuktikan anomali ini.

***Langkah percobaan :***

1. Lakukan trace packet terhadap tiga buah kondisi,
2. Kondisi pertama, trace packet pada saat melakukan login SSH secara normal, mulai dari konfirmasi pembuatan public-key sampai dengan autentikasi selesai dilakukan
3. Kondisi kedua, trace packet pada saat melakukan bruteforce attack pada SSH dengan satu username dan satu password yang sudah benar
4. Kondisi ketiga, trace packet pada saat melakukan brutefoce attack pada SSH dengan wordlist bersisi lima username dan lima password
5. Amati, perbedaan waktu yang dibutuhkan untuk masing masing percobaan bruteforce attack dan bandingkan dengan kondisi pada saat melakukan normal ssh.

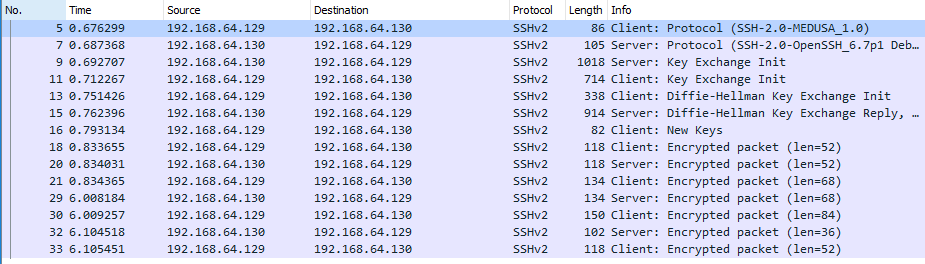
***Analisa :***

Packet trace pertama, yaitu normal login SSH ke server SSH dengan IP 192.168.64.130 dari IP 192.168.64.129.



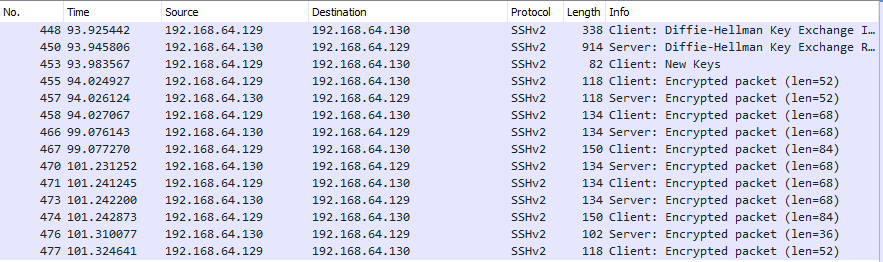
*Gambar 5.1 Login normal SSH membutuhkan waktu 12 detik untuk menyelesaikan prosesnya hingga autentikasi selesai*

Packet trace kedua, yaitu login SSH menggunakan medusa dengan user dan password yang sudah benar

**

*Gambar 5.2 Login menggunakan medusa satu password dan satu user membutuhkan waktu 6 detik untuk autentikasi*

Packet trace ketiga, yaitu login kombinasi lima user dan lima password SSH dengan menggunakan medusa.



*Gambar 5.3 Bruteorce menggunakan medusa dengan kombinasi 5 user dan 5 password, memakan waktu 101 detik.*

Pada gambar 5.3, bruteforce dengan medusa dilakukan dengan kombinasi 5 user dan 5 password yang berarti ada total 25 kombinasi user&password yang dicoba. Dengan hasilnya 25 kombinasi user&password tersebut dicoba oleh medusa dengan waktu total 101 detik. Sehingga, untuk menghitung berapa waktu yang dibutuhkan untuk melakukan mencoba satu kombinasi user&password adalah :

*101,324641 detik / 25 kombinasi = 4 detik/kombinasi*

Sehingga, waktu yang dibutuhkan medusa untuk melakukan satu kali penebakan password adalah 4 detik. Hal ini tentunya berlawanan dengan hasil percobaan packet pertama, dimana untuk melakukan login SSH normal membutuhkan waktu sampai dengan 12 detik.

Dari sinilah anomali yang terjadi pada bruteforce SSH ini dapat diamati. Jika normalnya adalah 12 detik, namun pada suatu waktu ada yang melakukannya hanya dengan waktu 4 detik, sudah dapat dipastikan bahwa percobaan login SSH tersebut adalah merupakan serangan bruteforce yang ditujukan ke SSH server tersebut.

Dengan acuan itu, dapat dikembangkan program dan aplikasi layaknya fail2ban untuk mengatasi/melindungi server terhadap serangan sejenis bruteforce attack SSH dengan medusa ini.