BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam keamanan jaringan ada bergbagai macam metode yang di gunakan untuk mengamankan jaringan internet. Keamanan jaringan internet tersebut disebut keamanan siber atau *Cyber Security*. Karena keamanan siber bukan hanya mengamankan jaringan internet tapi juga perangkat yang di gunakan. Keamanan siber membahas jenis ancaman, keamanan computer, keamanan internet, pr ivasi dan lainnya. Dalam penelitian ini salah satu jenis keamanan siber yang akan di bahas adalah Honeypot. Honeypot merupakan sebuah *software* keamanan jaringan yang bekerja dengan cara menciptakan sebuah *environment* baru yang menyerupai Server asli guna sebagai media untuk di serang oleh penyerang. Hal ini dapat membuat Server yang asli tidak terserang. Honeypot terbagi dua, *low-interaction honeypot* (LIH) dan *high-interaction honeypot* (HIH). Honeypot juga dapat membantu dalam mempelajari cara penyerang melakukan serangan karena di dalam Honeypot di bisa mengetahui apa saja yang di lakukan penyerang untuk melakukan serangan.

Honeypot itu sendiri telah perkembang cukup pesat dalam beberapa tahun terakhir ini. Dengan meningkatnya penggunaan *microservices* dan *containers* sebagai arsiktetur IT untuk teknoLogi modern maka orang mencoba mengalihkan penggunaan Honeypot kedalam *container* teresebut. Ada beberapa orang yang telah melakukan penelitian performa dan efisiensi Honeypot saat di jalankan pada *container* yaitu Docker. Salah satunya Dubravko Sever dan Tonimir Kišasondi. Dari hasil penelitian mereka penggunaan Honeypot di Docker memang meningkatkan efisiensi Honeypot. Itu didukung oleh karakteristik *container* yang tinggi efisiensi dan tinggi fleksibilitas. Yang lain yang mencoba meneliti Honeypot menggunakan *container* yaitu Andronikos Kyriakou dan Nicolas Sklavos. Hasil dari penelitan Andorinkos dan Nicolas hampir sama dengan hasil penilitan Dubravko dan Tonimir. Tapi dari hasil penelitan mereka ada 1 kekurangan yang ditemukan setelah mempelajari Honeypot. Yaitu efisiensi dalam dokumentasi dan montioring Log dari Honeypot tersebut. *Honeypot* menampilkan data Log lalu lintas jaringan yang di pantau dalam bentuk tampilan *command prom* atau terminal. Ada beberapa kesulitan yang di alami terutama jika di menggunakan

Server. UI yang kurang ramah untuk manusia, tidak ada detail jumlah Log yang masuk, tidak ada waktu dan tanggal Log tersebut.

Oleh karena itu pada penelitian ini ingin mencoba menggunakan aplikasi Elasticsearch dan Kibana untuk membuat hasil dari rekaman Honeypot tersebut lebih ramah bagi pengguna. Dengan menggunakan Elasticsearch sebagai mesin *database* dan Kibana sebagai aplikasi untuk menampilkan hasil dari Elasticsearch tersebut. Maka dapat membuat data hasil Honeypot lebih mudah untuk di baca dan dapat di simpan oleh pengguna, sehingga mampu mempermudah dalam menggunakan Honeypot tersebut.

1.2. Rumusan Masalah

Rumusan masalah dari penelitan ini adalah :

1. Bagaimana penerepan Elasticsearch agar dapat terhubung ke Honeypot.
2. Apakah data yang di terima oleh Elasticsearch 100% akurat sesuai dengan apa yang di rekam oleh Honeypot.
3. Apakah Elasticsearch mampu menerima data dalam jumlah besar yang di kirim dari Honeypot.

1.3. Tujuan dan Manfaat

Tujuan dari penelitian ini mengimplementasikan Elasticsearch agar dapat menampilkan Log Honeypot pada Kibana. Manfaat dari penelitian untuk mempermudah dalam memonitoring Log Honeypot pada Server.

1.4. Batasan Masalah

Batasan-batasan masalah pada penelitian ini yaitu :

1. Penelitian ini dilakukan menggunakan *virtual machine* (VM) dan aplikasi yang di gunakan adalah VirtualBox.
2. Di buat dalam bentuk 2 VM yaitu VM *Server* dan VM *Attacker*.
3. Honeypot di *install* pada *container* Docker dan di jalankan di Server.
4. Honeypot yang digunakan yaitu Honeytrap yang merupakan *low-interaction honeypot*.

1.5. Metode Penelitian

Dalam penelitian ini Langkah – Langkah atau metode dalam mengerjakan sebagai berikut:

1. Membuat sebuah simulasi virtual Server sederhana menggunakan VirtualBox, dimana berisi Server dan penyerang.
2. Menginstall Honeypot di Server sebagai media untuk mengelabui penyerang. Juga dapat membantu untuk mengumpulkan data.
3. Mengintall Kibana dan Elasticsearch untuk menerima semua data dari Honeypot.
4. Mencoba mengakses SSH yang telah di buat oleh Honeypot dari komputer penyerang dan melakukan be berapa jenis serangan untuk menguji Honeypot dan melihat apakah serangan serangan tersebut dapat dibaca oleh Kibana.