**Storyboard Client-Side Fipro**

Kami ingin membuat sebuah ***Dashboard*** untuk sistem informasi dan manajemen dari jaringan honeypot terdistribusi yang diantaranya memberikan akses pada *user* untuk melakukan manajemen sensor, memantau serangan, mengetahui informasi metric tertentu, mendapatkan informasi log serangan data sebuah file. Nantinya sebuah pengguna melakukan sebuah tindakan seperti berikut ini, dalam kondisi ketika ***main-server*** pada layanan jaringan honeypot terdistribusi ini telah aktif, sebuah pengguna menginginkan memasang sensor pada ip public tertentu yang telah tersedia.

Maka pengguna akan masuk pada fitur ***Deploy*** untuk mendapatkan sebuah link dengan command *wget* yang ditujukan untuk mendapatkan file bash script yang digunakan untuk melakukan instalasi sebuah web service pada sisi sensor yang digunakan sebagai webhook/listener/worker pada sisi sensor yang ditujukan pula untuk melakukan manajemen sensor. Ketika web service telah berhasil terinstalasi, pengguna dapat melakukan penambahan sensor honeypot melalui web client.

Untuk melakukan penambahan ***Sensor***, *user* harus masuk pada fitur sensor dan kemudian melakukan penambahan pada sensor, dalam hal ini satu jenis honeypot hanya dapat terinstal sekali pada setiap ***sensor-side-server***, pada fitur ini pula *user* dapat mengetahui perubahan informasi pada setiap sensor honeypot dan juga dapat mengakses secara langsung untuk log pada sensor honeypot yg bersangkutan. Fitur ini pula juga menyediakan untuk melakukan penghapusan sensor honeypot pada *sensor-side-server*.

Kemudian terdapat fitur ***Log***, fitur log ditujukan untuk melakukan akses secara langsung pada log seperti log untuk sensor cowrie yang bersifat sequential, untuk sensor dionaea ditujukan dengan informasi malware yang bersangkutan karena nantinya malware dalam format binary ini akan didownload dari sensor-side-server menuju ***main-server*** untuk kepentingan lebih lanjut, pada sensor glastopf memberikan informasi dorks file pada setiap log yg berhasil didapatkan. Dan yang terakhir adalah memungkinkan pengguna untuk mendownload log file dalam format csv.

Terdapat fitur ***Maps Attack***, fitur ini ditujukan untuk visualisasi attacks pada map yang merujuk pada existing system dari saudara Yasin, fitur maps akan menampilkan visualisasi serangan secara realtime maupun historical.

Sehingga untuk versi pertama dari Fipro akan terdapat 5 jenis fitur utama yang tersedia pada *client-side* yang ditujukan pada pengguna.

Dalam perkembangannya nanti, akan diberikan sebuah fitur yang digunakan untuk tujuan sharing data dari setiap sensor pengguna yang berbeda, dengan ketentuan pengguna memberikan ***privilege public*** pada log file yang dihasilkan oleh sensor honeypot mereka.

**Storyboard sistem keseluruhan**

Dalam membangun jaringan honeypot terdistribusi berbasis container ini, kami mengkombinasikan penggunaan ***bash-scripting***yang ditujukan untuk kemudahan instalasi sistem dan penggunaan ***container-based*** yang ditujukan untuk fleksibilitas dan keamanan sensor.

Bash Scripting yang dibangun antara lain memiliki kemampuan sebagai berikut:

* **instal**.sh : ditujukan untuk melakukan instalasi ***main-server*** dengan segala perabotannya. ( instalasi webservice, docker thing, mqtt service, maxmind, membuat directory untuk malware binary)
* **install\_sensor**.sh : ditujukan untuk melakukan instalasi ***sensor-side-server*** dengan segala perabotannya. (membuat directory ‘/data’, ‘new user’, instalasi docker)
* **install\_wssensor**.sh : ditujukan untuk melakukan instalasi web service pada ***sensor-side-server*,** yang digunakan sebagai webhook/listener untuk setiap request yang dikirim pada server dalam melakukan manajemen sensor honeypot.
* **install\_docker**.sh : digunakan untuk melakukan proses instalasi docker engine dan client
* **install\_dockercompose**.sh : digunakan untuk melakukan proses instalasi docker compose pada ***main****-****server***
* **set\_sensorconf**.sh : digunakan untuk merubah beberapa variable yang telah ditentukan pada setiap komponen yang berkaitan dengan ***sensor-side-server***

Container yang dibangun:

* dionaea honeypot (**sensor**) as sensor
* cowrie honeypot (**sensor**) as sensor
* glastopf honeypot (**sensor**) as sensor
* fluentbit (**sensor**) as log forwarder
* ws-sensor (**sensor**) as webhook
* fluentd-custom-with-plugin-mqtt+record\_modifier (**server**) as log aggregator
* ws-server (**server**) as main server
* mqtt (**server**) as data collector
* mongodb (**server**) as main database

Dalam proses pembangunan ini terdapat sebuah evaluasi selama proses testing pada periode bulan januari - Februari 2018, bahwasannya sensor mengalami stuck proses karena seluruh komponen seperti CPU Usage, Memory Usage, Storage Usage mengalami 100% penggunaan pada hamper keseluruhan yang mengakibatkan machine tidak dapat diakses / sangat amat pelan.

Sehingga dari kasus tersebut didapatkan evaluasi bahwa harusnya terdapat sebuah **cronjob** untuk melakukan rotasi pada log dengan **logrotate** yang bertujuan untuk tetap menghapus secara berkala dalam hal ini kami menentukan penghapusan dilakukan setiap 24 jam sensor berjalan.

**Storyboard Testing/Uji Sistem Fipro**

Dalam melakukan uji testing pada sistem kami, tentu yang utama adalah melakukan unit testing pada seluruh komponen program (optional).

Kemudian dalam membangun scenario testing kami mempersiapkan terdapat 2 skenario testing yang dapat diuji dalam penggunaan sistem kami diantaranya adalah:

* Bandwith Testing (Digunakan untuk mengetahui kemungkinan bandwith yang digunakan selama proses data masuk sampai keluar dan diterima oleh ***main-server***)
* Application Programming Interface Testing (Digunakan untuk melakukan pembuktian bahwasannya sistem kami telah berjalan dengan mulus sesuai ekspektasi fungsi kerja API)
* Usage Testing (Digunakan untuk menguji sistem kami berjalan dalam jangka waktu tertentu dan memvisualisasikan serangan secara baik pada dashboard untuk ***client-side* user**)
* Usage Testing #2 ( Digunakan dengan menguji beberapa ***sensor*** yang tersebar pada beberapa server dalam hal ini sesuai yang diekspektasikan adalah sebanyak 10 server dengan maksimal sensor sebanyak 30 sensor)
* Selesai