

INTISARI

Sistem *Monitoring* Performa Raspberry Pi dan *Honeypot* pada Penggunaan Sumber Daya Berbasis Elastic Stack dengan Notifikasi Telegram

Berdasarkan penelitian oleh P. Dilsheer Ali et al. (2017) mengatakan setiap harinya serangan dunia maya semakin pesat sehingga perlu dibentuk mekanisme keamanan yang semakin ketat dan modern. Sebagai improvisasi dari teknologi perlindungan serangan keamanan, saat ini berbagai organisasi atau perusahaan sudah beberapa menerapkan *honeypot*. Akan tetapi permasalahan yang ditemukan baru-baru ini yaitu pada kinerja implementasi *honeypot* tersebut, apakah *honeypot* diketahui aktif atau tidak dan bagaimana konsumsi sumber daya *honeypot* yang dapat memengaruhi *honeypot* maupun perangkat yang ditumpangi. Penelitian ini mengusulkan implementasi sistem *monitoring* performa *honeypot* pada penggunaan sumber daya yang divisualisasikan dengan Kibana dan terintegrasi dengan notifikasi Telegram sebagai kemudahan administrator dalam mengelola *resource* dan status keaktifan *honeypot* maupun Raspberry Pi. Selain itu, apabila *honeypot* diketahui tidak aktif maka dilakukan otomatisasi mengaktifasi *honeypot* secara jarak jauh dari perangkat Raspberry Pi yang terhubung pada *server* menggunakan bahasa pemrograman python. Pengujian beban dilakukan untuk mengidentifikasi performa *honeypot* berdasarkan sumber daya. Berdasarkan pengujian beban pada Dionaea dihasilkan kenaikan *storage* sebesar 0,4% dari 1000 beban, 1,2% dari 3000 beban, dan 1,91% dari 5000 beban. Persentase kenaikan memori sebesar 31,94% dari 1000 beban, 37,3% dari 3000 beban, dan 40,24% dari 5000 beban.

Kata kunci : *monitoring*, *honeypot*, Raspberry Pi, Elastic Stack, *load testing*, *python*

ABSTRACT

Based on research by P. Dilsheer Ali et al. (2017) said that cyber attacks are increasing rapidly every day, so it is necessary to establish a more stringent and modern security mechanism. As an improvisation of security attack protection technology, currently various organizations or companies have implemented honeypots. However, the problems that were discovered recently are the performance of the honeypot implementation, whether the honeypot is known to be active or not and how the consumption of honeypot resources can affect the honeypot and the device it hosts. This study proposes the implementation of a honeypot performance monitoring system on the use of resources that is visualized with Kibana and integrated with Telegram notifications as an administrator's convenience in managing resources and the active status of the honeypot and Raspberry Pi. In addition, if the honeypot is found to be inactive, then automation will activate the honeypot remotely from the Raspberry Pi device connected to the server using the python programming language. Load testing is performed to identify honeypot performance based on resources. Based on the load test on Dionaea, storage increased by 0.4% from 1000 loads, 1.2% from 3000 loads, and 1.91% from 5000 loads. The percentage increase in memory is 31.94% for 1000 loads, 37.3% for 3000 loads, and 40.24% for 5000 loads.

Keywords : monitoring, honeypot, Raspberry Pi, Elastic Stack, load testing, python