**semanTIK**, Vol.7, No.1, Jan-Jun 2021, pp. 17-26
ISSN: 2502-8928 (Online)
DOI: 10.5281/zenodo.5034918
**Terakreditasi "Peringkat 4 (Sinta 4)" oleh Kemenristekdikti**
Website: http://ojs.uho.ac.id/index.php/semantik



# IMPLEMENTASI *HONEYPOT* DAN *PORT KNOCKING* DALAM MENDETEKSI SERANGAN *DDoS ATTACK* PADA SERVER JARINGAN

**Suliman[*1], Andani Achmad[2], Adnan[3]**
[1]Program Studi Sistem Komputer STMIK Bina Bangsa Kendari
[2]Program Studi Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin
[3]Departemen Teknik Informatika, Universitas Hasanuddin
e-mail: [*1]**suliman170892@gmail.com**, [2]andani@unhas.ac.id, [3]adnan@unhas.ac.id

**Abstrak**

Sistem keamanan jaringan semakin hari kian makin berkembang, begitu pula serangan pada sistem jaringan yang berbeda-beda metode dan perkembangannya, khususnya pada *server* yang menjadi pengendali utama dalam sistem jaringan menjadi target utama. Oleh karena itu pentingnya menggunakan sistem keamanan jaringan dalam mendeteksi dan menggagalkan serangan.

Metode serangan yang digunakan adalah *Distributed Denial of Service* (*DDoS attack*) dengan berbagai jenis serangan seperti *DDoS attack request flooding*, *DDoS attack traffick flooding*, *DDoS brute force attack* dan *DDoS attack SQL injection.* Sedangkan untuk mendeteksi dan menggagalkan serangan yaitu menggunakan metode *honeypot* dan *port knocking* yang akan berkolaborasi dalam mengamankan sistem *server* jaringan pada sistem operasi Windows. Hasil saat dilakukan 5 kali proses pengujian sebelum serangan rata-rata *performance* kinerja *CPU usage history* (5%-18%) dan *networking LAC* (1%-5%). Setelah serangan, *CPU usage history* (72%-90%) dan *networking LAC* (22%-32%). Sedangkan pada saat ada serangan namun *server* jaringan sudah terpasang *honeypot* dan *port knocking performance* kinerja *CPU usage history* menjadi stabil mencapai rata-rata (5%-21%) dan *networking LAC* (1%-6%).

**Kata kunci**; *Server Jaringan*, *DDoS Attack*, *Honeypot* dan *Port Knocking*

***Abstract***

*The network security system is growing day by day, as are attacks on network systems with different methods and developments, especially on servers that are the main controllers in the network system, which are the main targets. Therefore, the importance of using a network security system in detecting and thwarting attacks.*

*The attack method used is a Distributed Denial of Service (DDoS attack) with various types of attacks such as DDoS attack request flooding, DDoS attack traffick flooding, DDoS brute force attack and DDoS SQL injection attack. Meanwhile, to detect and thwart attacks, namely using honeypot and port knocking methods that will collaborate in securing network server systems on Windows operating systems. The results when doing 5 times the testing process before the attack, the average performance performance of CPU usage history (5% -18%) and networking LAC (1% -5%). After CPU usage history attack (72% -90%) and Networking LAC (22% -32%) while during an attack, the network server has a honeypot installed and the port knocking performance, the CPU usage history performance is stable, reaching an average (5% -21%) and networking LAC (1% -6%).*

***Keywords***; *Network Servers, DDoS Attack, Honeypot and Port Knocking*

## 1. PENDAHULUAN

Sebelum adanya metode *honeypot* dan *port knocking* untuk keamanan *server* pada sistem operasi biasanya komputer *server* hanya menggunakan *firewall* untuk mengamankan lalu lintas jaringan, namun *firewall* pun masih banyak memiliki kelemahan dan kekurangan. Untuk mengatasinya maka dibutuhkan pengembangan dari *firewall* yaitu dengan mengimplementasikan *honeypot* dan *port knocking* pada *server*. Dimana *port knocking* dapat mengontrol layanan *port* terbuka dan *port* tertutup [1]. Selain menggunakan metode *port knocking* dibutuhkan *honeypot* untuk mengalihkan *attacker* ke dalam *server* tiruan dan mendeteksi serangan apa saja yang dilakukan oleh *attacker/intruder* pada *server* sehingga penyerang terjebak dan tidak mengganggu *server* pada jaringan [2].

Pengujian penelitian ini dilakukan pada Laboratorium Kampus Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Bina Bangsa Kendari yang dimana setiap proses ujian kompetensi mahasiswa dilakukan secara *online* maupun *offline* untuk lebih mempermudah dalam proses ujian dan akurat dalam memperoleh hasil nilai, namun pada pengelolaan jaringan hanya menggunakan satu *server* untuk melayani *user client* yang nantinya akan digunakan oleh mahasiswa dalam melakukan proses ujian, yang tentunya membutuhkan sistem keamanan jaringan yang baik dan handal untuk mengantisipasi jika terjadi adanya gangguan pada *server* jaringan dari serangan *user* yang tidak bertanggung jawab dengan tujuan mengganggu server jaringan dengan metode serangan *DDoS Attack* untuk melumpuhkan *client-server* agar tidak bisa terhubung pada jaringan atau jaringan menjadi *down*, eksploitasi *password* dan username pada *microtik router* dan menginjeksi *database* pada aplikasi web untuk bisa mendapatkan *password* dan *username* sehingga dapat login sebagai admin. Model sistem jaringan yang akan dibangun pada pengujian penelitian ini yang nantinya akan dijadikan sebagai target serangan dan pengujian dalam keamanan *server* jaringan adalah simulasi ujian offline jaminan internal mutu (JAMIN) dimana ujian ini biasanya digunakan pada mahasiswa semester VI (enam) dengan tujuan untuk menguji mahasiswa tentang kemampuannya dalam bidang ekstrakurikuler dasar computer. Aplikasi web *Computer Based Test* (CBT) digunakan pada calon mahasiswa baru untuk melakukan tes Seleksi Bersama Masuk Perguruan Tinggi Swasta (SBMPTS) dan *router microtik* yang nantinya akan digunakan untuk mengatur konvigurasi *server* pada jaringan. Sedangkan untuk keamanan *server* jaringan yaitu menggunakan metode *Honeypot* dan *Port Knocking* dengan tujuan mengamankan server jaringan untuk menjebak dan memblokir serangan dari user yang tidak bertanggung jawab dari serangan *DDoS attack* pada sistem operasi Windows.

Penelitian sebelumnya pada jurnal [3] pengujian dilakukan menggunakan *server Ubuntu* untuk keamanan *server* dengan *honeypot* dan *port knocking*, teknik serangan menggunakan aplikasi *MobaXtreme* dan *Putty*. Penelitian yang lain [4] hanya menggunakan metode *honeypot* dalam keamanan *server* dikonfigurasi menggunakan sistem operasi *Linux*, teknik serangan *DDoS attack* dengan aplikasi *loic*. Pada jurnal [5] *Honeypot* ini berjenis *low interaction* menggunakan bahasa pemrograman *python* dalam konsep *network programming* untuk mengimplementasi *protokol* SSH yang digunakan untuk memantau *server*.[6] Mengamankan *port* pada sistem *server* jaringan pada sistem operasi *linux* menggunakan metode *port knocking*. [7] Keamanan sistem jaringan secara *realtime* pada *Ubuntu server* juga *implementasi* fail2ban pada *Ubuntu server* versi 16 untuk mencegah serangan *brute force* dan *DDOS*.

## 2. METODE PENELITIAN

### 2.1 *Honeypot*

*Honeypot* adalah sistem umpan untuk mengumpulkan informasi *attacker* dari penyerang dengan cara menunggu, memantau setiap aktivitas *attacker* yang memulai interaksi, mengumpulkan sebanyak–banyaknya data yang dikenali sebagai sebuah serangan untuk melindungi sistem dan jaringan [8].

*Honeypot* tidak bisa mencegah serangan *cyber* terhadap jaringan sendiri, tetapi mereka dapat membantu dalam mengidentifikasi dan melakukan deteksi terhadap serangan ketika mereka digunakan bersama dengan perangkat pertahanan lainnya seperti *firewall*. *Honeypot*

dapat menghasilkan sejumlah data yang memiliki nilai yang tinggi dan dapat juga menjadi tantangan bagi para pakar keamanan professional [9].

2.2 *Port Knocking*

*Port Knocking* adalah sebuah metode sederhana untuk memberikan akses remot tanpa meninggalkan *port* dalam keadaan selalu terbuka. Hal ini akan memberikan perlindungan kepada *server* dari *port scanning* dan *serangan scripts kiddie*s [10].

*Port Knocking* memiliki metode membuka *port* kepada suatu *klient* bila *client* itu meminta dan ditutup kembali jika *client* telah selesai. Untuk menjalankan metode ini, sebuah *server* haruslah memiliki *firewall* untuk menjalankan metode *port knocking* yang berjalan di *server* tersebut. *User* dituntut untuk memberikan autentikasi ke *server* agar *firewall* menulis ulang rulenya sehingga *user* diberi izin untuk mengakses *port* yang dimaksud. Setelah selesai, *user* mengirimkan autentikasi kembali untuk menutup *port* agar *firewall* menghapus rule yang ditulis sebelumnya untuk membuka *port.*

2.3 *DDoS Attack* (*Distributed Denial of Service Attack*)

*DDoS* merupakan teknik serangan yang paling popular [11]. *DDoS* merupakan salah satu ancaman utama dunia maya dan menjadi masalah utama keamanan *cyber*. Serangan *DDoS* mengalami perkembangan teknik yang mutakhir sebagai contoh adalah serangan *SYN-Flood*. Pada sistem *SYN-Flodd alert* bersifat *false-positive* yang sering terjadi pada IDS yang berbasis *signature*, dimana pola jaringan normal dideteksi sebagai serangan *DDoS*, sehingga ketika benar-benar terjadi serangan *DDoS* waktu untuk menentukan dan melakukan tindakan mitigasi secara cepat untuk mengamankan jaringan tidak bisa dilaksanakan seefisien mungkin [12].

Pada proses pengujian serangan dalam penelitian ini ada beberapa teknik *DDoS attack* yang akan digunakan, yang nantinya akan dianalisis bagaimana keadaan komputer *server* pada jaringan sebelum dan sesudah adanya serangan dan bagaimana ketika sudah diterapkannya implementasi dari kemanan jaringan dengan menggunakan *honeypot* dan *port knocking* yaitu sebagai berikut;

1. Serangan *DDoS Attack Request Flooding*

*Request Flooding* adalah sebuah teknik serangan *DDoS* yang melakukan pengiriman *request* secara terus menerus sehingga membanjiri lalu lintas jaringan. Akibatnya pengguna lain yang juga meminta layanan tidak dapat dilayani oleh *server*.

2. Serangan *DDoS Attack Traffic Flooding*

*Traffic Flooding* merupakan teknik yang sistem kerjanya sama seperti *request flooding*. Pembedanya adalah paket yang dikirimkan, jika dalam *request flooding* yang dikirim adalah *request* sedangkan dalam teknik *traffic flooding* yang dikirim adalah *byte* data sehingga pengguna lain tidak dapat dilayani.

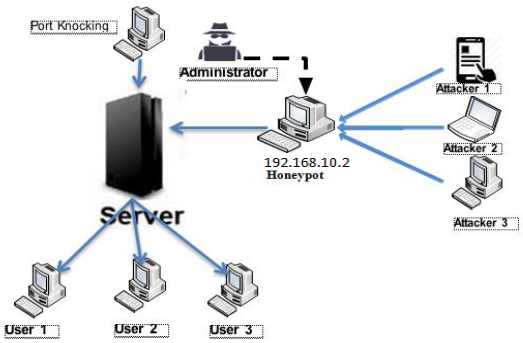
3. Serangan *DDoS Brute Force attack*

Teknik *hacking Brute Force* adalah salah satu teknik penyerang untuk meretas *password* sebuah *server*, jaringan atau *host*, dengan cara mencoba semua kemungkinan kombinasi *password* yang ada pada *wordlist* atau "kamus *password*". Pada penelitian ini penyerang akan menggunakan metode *exploit microtik* dengan *winbox exploit* untuk mendapatkan *password microtik* agar bisa masuk kedalam *server* jaringan.

4. Serangan *DDoS attack SQL Injection*

SQL digunakan untuk melakukan *query*, mengoperasikan dan mengelola sistem *database* seperti SQL *server*, *oracle*, atau MySQL. Penggunaan umum SQL konsisten di semua sistem *database*, namun ada detail perbedaan tertentu yang khusus untuk setiap sistem. *Structured Query Language* (*SQL) injection* adalah jenis aksi *hacking* pada keamanan komputer di mana seorang penyerang bisa mendapatkan akses ke basis data di dalam sistem dengan memanfaatkan sebuah celah keamanan yang terjadi dalam lapisan basis data untuk mendapatkan *password* atau *username* pada sebuah aplikasi web yang tidak diproteksi dengan baik.

Sebelum serangan *DDoS attack* dilakukan terlebih dahulu penyerang melakukan *scanning ip addres* dan *port* yang digunakan oleh *server* jaringan dengan menggunakan *software advanced ip scanner* dan *Scaning port nmap* setelah *ip* dan *port* ditemukan barulah pengujian serangan *DDoS attack* dilakukan.




Gambar 1. Gambaran umum implementasi *honeypot* dan *prot knocking* pada *server* jaringan

Pada gambaran umum sistem *honeypot* dan *port knocking* pada Gambar 1, *server* jaringan menjadi target dari serangan *DDoS attack*, namun serangan tersebut telah dialihkan dan masuk kedalam *server honeypot* yang dimonitor langsung oleh administrator, yang selanjutnya administrator akan mengambil langkah-langkah tindakan pemblokiran *user* serangan *DDoS attack* dengan metode *port knocking*.

## 3. HASIL DAN PEMBAHASAN

Hasil penelitian yang didapatkan dari pengujian terhadap sistem keamanan *server* jaringan menggunakan *honeypot* dan *port knocking* dari serangan *Distributed Denial of Servis Attack* (*DDoS Attack*) adalah sebagai berikut;

### 3.1 Simulasi serangan *Distributed Denial of Servis Attack (DDoS Attack)* sebelum menggunakan *Honeypot* dan *Port Knocking*

Sebelum serangan dilakukan penyerang harus mendapatkan *ip* target terlebih dahulu agar bisa melakukan serangan ke sistem *server* jaringan yang digunakan untuk bisa melakukan *DDoS attack*, *tools* yang akan digunakan yaitu *advanced ip scanner*.

Tabel 1. Hasil scanning ip address dan port

| No | Jenis Scanning | Hasil ip address | Hasil port |
|---|---|---|---|
| 1 | *Advanced ip scanner* | 192.168.10.1<br>192.168.10.2<br>192.168.10.4<br>192.168.10.11 | - |
| 2 | *Scaning port nmap* | - | 80<br>21 |

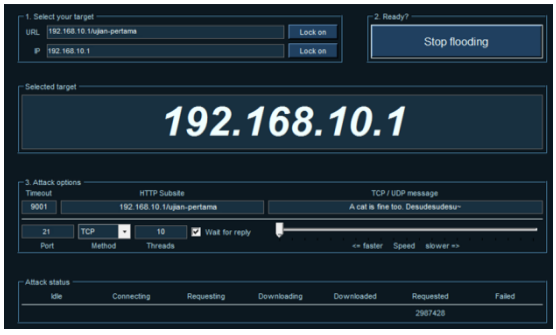
Setelah didapatkan hasil *scanning ip address* dan *port* target, dimana *ip* yang digunakan *server* adalah 192.168.10.1 terdapat keterangan pada *manufactur* adalah *routerboard* dan *port* 21 terdapat keterangan pada *service* yaitu *microsoft-ds.* Tahapan selanjutnya adalah melakukan serangan *DDoS attack* dimana pada *server* jaringan ini belum dikonfigurasi dengan *honeypot* dan *port knocking*, berikut tahapan-tahapan dan hasil serangan *Distributed Denial of Servis attack* (*DDoS attack*).

#### 3.1.1 Serangan *DDoS attack* pada aplikasi ujian jamin berbasis *offline*

Pada tahap ini pengujian penelitian yang dilakukan yaitu menggunakan jaringan *Local Area Network* dengan simulasi ujian berbasis *offline* dengan percobaan pengujian serangan menggunakan 2 *software DDoS* yaitu *hoic* untuk *request flooding* pada jaringan dan *loic* untuk *traffic flooding* pada jaringan secara bersamaan untuk mengirim *request* dan paket data dalam jumlah banyak sehingga CPU pada komputer *server* bekerja semakin keras dan lalulintas jaringan menjadi sibuk dan padat sehingga jaringan yang terhubung ke *server* sangat lambat dan bahkan sebagian tidak bisa untuk diakses.

1. Serangan *DDoS Attack Request Flooding*

Pada pengujian *DDoS Attack Request Flooding* untuk mengirimkan *reques*t pada jaringan secara terus menerus penyerang menggunakan *software loic* tujuannya adalah untuk membanjiri lalu lintas jaringan sehingga *server* jaringan tidak bisa melayani pengguna lain. Setelah *software* berjalan masukan alamat *URL* target 192.168.10.1/ujian-pertama dan *ip* target 192.168.10.1 yang sudah terdeteksi.



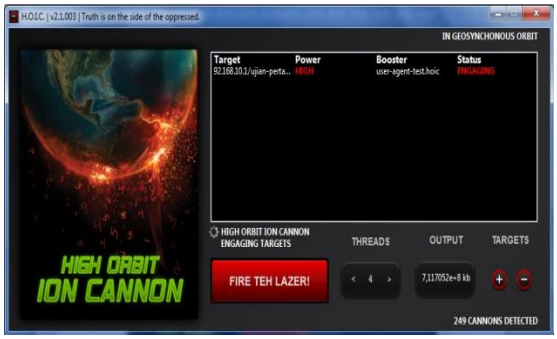

Gambar 2. *DDoS attack request flooding*

*Request flooding* sedang berjalan pada *port* 21, *metdhod* TCP dan *threads* 10 dengan level kecepatan *faster* atau kecepatan tertinggi dalam level pengujian ini, *name* serangan

*desudesudesu* pengujian serangan akan dihentikan jika sistem jaringan sudah *down*.

2. Serangan *DDoS Attack Traffic Flooding*

Pada serangan *DDoS attack hoic* masukan alamat *URL* dari halaman *website* yang terhubung ke jaringan untuk selanjutnya menjadi target penyerang memasukan *bytes* data dalam jumlah banyak pada pengujianya menggunakan *URL* 192.168.10.1/ujian-pertama sebagai target *server*, sedangkan untuk *power* pilihanya ada *low*, *medium* dan *high* peneliti menggunakan *high* untuk kecepatan tertinggi, dalam penelitian ini penyerang akan menggunakan *user-agent-test.hoic* untuk mengirim paket data ke *server* jaringan.




Gambar 3. *DDoS attack traffic flooding*

Status pada *software hoic ENGAGING* yang artinya sedang berjalan untuk memasukan paket *traffic* berupa *byte* data kedalam *ip server* sebagai target untuk mengganggu sistem jaringan agar tidak bisa digunakan antara *user* komputer *client-server*. Dari hasil pengujian serangan pada aplikasi web ujian jamin berbasis *offline* dengan model jaringan *localhost* pada *server* jaringan menggunakan *DDoS attack request flooding* dan *traffick flooding* serangan berhasil masuk pada komputer server jaringan berupa *request* paket dan *byte* data.

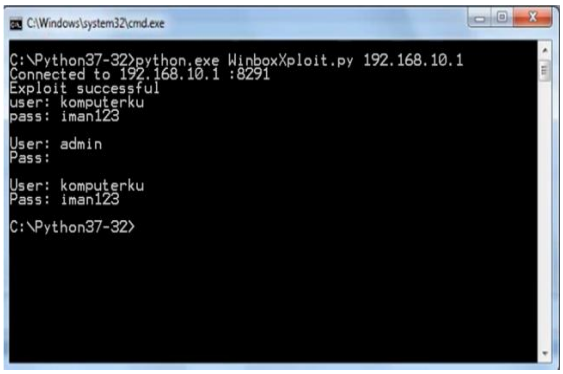
Tabel 2. Hasil serangan *DDoS attack request flooding* dan *DDoS attack traffick flooding*

| Pengujian | Ip target/ Port | Waktu | Hasil serangan |
|---|---|---|---|
| **Uji 1** | *DDoS attack request flooding* 192.168.10.1/21 | 31 menit | 2987428 *seconds* |
| | *DDoS attack traffick flooding* 192.168.10.1/21 | 31 menit | 7.117052e+8 kb |
| **Uji 2** | *DDoS attack request flooding* 192.168.10.1/21 | 25 Menit | 2735891 *seconds* |
| | *DDoS attack traffick flooding* 192.168.10.1/21 | 25 menit | 5.363787e+8 kb |
| **Uji 3** | *DDoS attack request flooding* 192.168.10.1/21 | 22 menit | 2583947 *seconds* |
| | *DDoS attack traffick flooding* 192.168.10.1/21 | 22 menit | 7.023877e+7 kb |
| **Uji 4** | *DDoS attack request flooding* 192.168.10.1/21 | 37 menit | 3310978 *seconds* |
| | *DDoS attack traffick flooding* 192.168.10.1/21 | 37 menit | 2.116352e+9 kb |
| **Uji 5** | *DDoS attack request flooding* 192.168.10.1/21 | 29 menit | 2872984 *seconds* |
| | *DDoS attack traffick flooding* 192.168.10.1/21 | 29 menit | 1.080268e+9 kb |

Pada hasil pengujian serangan jaringan sebelum digunakannya *honeypot* dan *port knocking* didapatkan sistem jaringan menjadi tidak stabil halaman web yang terhubung ke *server* jaringan pun menjadi *down* akibat banyaknya paket *byte* data dan *request* yang masuk pada *server* jaringan.

3.1.2 Serangan *DDoS Brute Force Attack*

Pada penelitian ini *soffware* yang digunakan adalah *phyton* versi 3.8.5 dan dikonfigurasikan dengan *script winbox exploit*. Setelah selesai menginstal *phyton* selanjutnya *ekstrak* file *script winbox* kedalam folder *phyton* lalu buka *software winbox* dan masukan *ip address microtik* target serangan.




Gambar 4. Hasil *exploit password microtik*

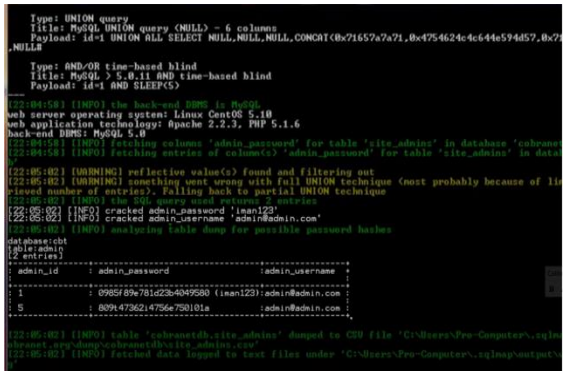
Pada proses *exploit password* dan *user* didapatkan *password* = iman123 sedangkan *user* = komputerku berhasil terdeteksi yang selanjutnya akan kita coba *login* pada *microtik* menggunakan *winbox*.

Tabel 3. Hasil serangan *DDoS brute force attack*

| NO | Target serangan | Status serangan | Hasil serangan |
|---|---|---|---|
| Uji 1 | FTP *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| | SSH *brute* | Router | Sukses |

| | | | |
|---|---|---|---|
| | *force attack* | memproduksi log | menampilkan *username* dan *password router microtik* |
| Uji 2 | FTP *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| | SSH *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| Uji 3 | FTP *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| | SSH *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| Uji 4 | FTP *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| | SSH *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| Uji 5 | FTP *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |
| | SSH *brute force attack* | Router memproduksi log | Sukses menampilkan *username* dan *password router microtik* |

### 3.1.3 Serangan *DDoS Attack SQL Injection*




Gambar 5. *password* dan *username admin*

Tahap terakhir didapatkan hasil *password* dan *username* dari aplikasi web tersebut dengan *password* 'iman123' sedangkan *username* 'admin@admin.com' yang selanjutnya akan digunakan penyerang untuk dapat masuk kedalam aplikasi.

Tabel 4. Hasil *SQL Injection*

| NO | Jenis *eksploitasi* | Hasil *SQL Injection* | | Keterangan |
|---|---|---|---|---|
| | | Berhasil | Gagal | |
| 1 | Database | √ | - | Terdapat 3 database yaitu cbt, information_schema dan test |
| 2 | Table | √ | - | Terdapat 14 table yaitu table dosen, groups, h_ujian, jurusan, jurusan_matkul, kelas, kelas_dosen, login_attempts, mahasiswa, matkul, m_ujian, tb_soal, admin dan users_group |
| 3 | Column | √ | - | Terdapat ada 3 yaitu admin_id, admin_passwword dan admin_username |
| 4 | Entries | √ | - | Terdapat 2 entries yaitu *password* (iman123) dan *username* (admin@admin.com) |

Dari hasil *exploitasi database* yang dilakukan oleh penyerang menggunakan teknik *SQL Injection sqlmap* terdapat beberapa struktur *database*, *table* dan *coloumn* yang ditampilkan yang selanjutnya didapatkan *password* dan *username admin* yang akan digunakan untuk masuk kedalam aplikasi web tersebut setelah tervalidasi dan dicoba masuk ternyata penyerang berhasil masuk kedalam aplikasi web dengan menggunakan *username admin.*

Tabel 5. Hasil pengujian jaringan sebelum menggunakan *honeypot* dan *port knocking*

| NO | Jenis serangan | Proses *runing* | | Keterangan |
|---|---|---|---|---|
| | | Berhasil | Gagal | |
| 1 | *DDoS Attack request Flooding* | √ | - | *Request* paket masuk setiap *and off conection* 2987428 *seconds*, 2735891 *seconds*, 2583947 *seconds*, 3310978 *seconds* dan 2872984 *seconds*. |
| 2 | *DDoS Attack Traffic Flooding* | √ | - | *Traffick* paket data yang masuk 7.117052e+8 kb, 5.363787e+8 kb, 7.023877e+7 kb, 2.116352e+9 kb dan 1.080268e+9. |
| 3 | *DDoS Brute Force Attack* | √ | - | Berhasil menampilkan *username* dan *password router microtik* |
| 4 | *DDoS Attack SQL* | √ | - | Berhasil menampilkan *password* (iman123) |

| | | | | |
|---|---|---|---|---|
| | *injection* | | | dan *username* (admin@admin.com ) pada aplikasi web |

3.2 Konfigurasi *Honeypot* dan *Port Knocking*

Tabel 6. Hasil konfigurasi *honeypot* dan *port knocking* pada *server* jaringan

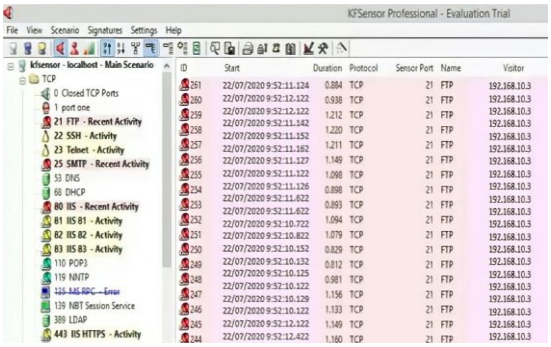
| Jenis keamanan | Hasil konfigurasi | | Rule konfigurasi | Keterangan |
|---|---|---|---|---|
| | Berhasil | Gagal | | |
| *Honey pot* | √ | - | Protocol TCP, port FTP 21, *ip* 192.168.10 .2 | Mengalihkan/ menjebak serangan *DDoS attack* dan merekam log aktifitas pada server jaringan |
| *Port Knocking* | √ | - | Port webfig (tcp 80), SSH (tcp 22), winbox (tcp 8291), FTP (tcp 21), *ip* 192.168.10 .1 | Mengamankan port pada server jaringan |

Konfigurasi pada *server honeypot* menggunakan *ip* 192.168.10.2 sebagai alamat untuk menjebak serangan *DDoS attack, port* yang digunakan adalah FTP 21 *protocol* TCP karena sistem jaringan ini menggunakan *localhost area connection*, sedangkan *port knocking* bertugas mengamankan *port-port* pada jaringan agar tidak mudah bagi *user* asing masuk kedalam sistem jaringan dan untuk memblokir *user* yang tidak dikenal jika ingin mengganggu sistem jaringan, *port-port* yang diamankan adalah *port* 80,22, 8291 dan 21.

3.3 Analisis serangan setelah menggunakan keamanan jaringan *Honeypot* dan *Port Knocking*

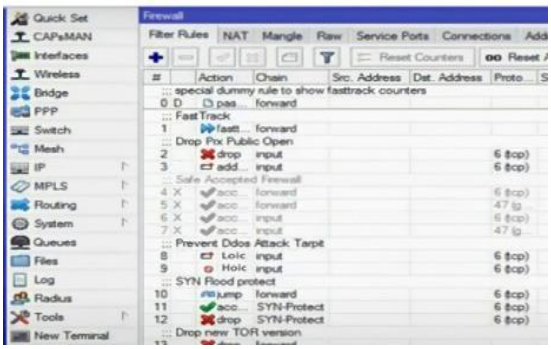
1. Mendeteksi serangan DDoS *attack request flooding* dan *traffick flooding*

Setelah *honeypot* terpasang pada *sever* jaringan dan *port knocking* telah diatur konfigurasinya pada *microtik* untuk keamanan *server* jaringan, selanjutnya akan dilakukan penyerangan ulang agar dapat menganalisis hasil dari serangan setelah *honeypot* dan *port knocking* terpasang pada sistem jaringan *server*. Dalam penelitian ini penyerang menggunakan *ip* 192.168.10.3 untuk melakukan serangan ke *ip server* jaringan.



Gambar 6. Honeypot mendeteksi serangan *DDoS attack request flooding* dan *traffick flooding*

Serangan *DDoS attack* yang masuk pada sistem jaringan berhasil dialihkan oleh *honeypot KFSensor* yang bekerja secara *realtime.* sehingga serangan tidak sempat merusak pengguna lain yang sedang terhubung ke jaringan server untuk melakukan ujian jamin *offline*, analisa serangan yang masuk yaitu dari *port* 21 FTP dengan *ip address* 192.168.10.3. Pengujian serangan dilakukan dengan lama waktu rata-rata 10 menit karena setelah serangan *DDoS attack request flooding* dan *traffick flooding* terdeteksi oleh *honeypot* selanjutnya dilakukan pemblokiran *user* dengan *port knocking.*



Gambar 7. Memblokir serangan *DDoS attack* pada *port knocking*

Setelah masuk pada *microtik* dengan metode *port knocking* didapatkan telah masuk serangan DDoS *attack request flooding* dengan menggunakan *software loic* dan *traffick flooding* menggunakan *software hoic* yang selanjutnya akan dilakukan pemblokiran pada *address list* agar serangan benar-benar dihentikan sehingga tidak lagi mengganggu sistem pada *server* jaringan.

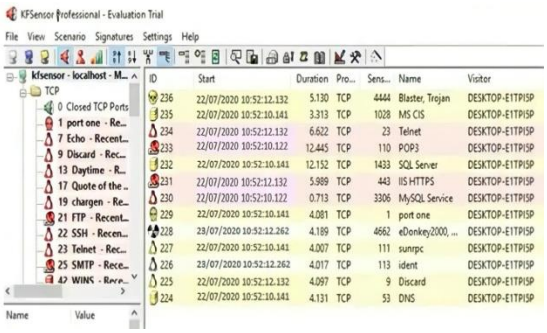
Tabel 7. Analisis hasil komputer *server* setelah terpasang *honeypot* dan *port knocking*

| Pengujian | Rata-rata performance | Sebelum serangan | Setelah serangan | Setelah terpasang *honeypot* dan *port knocking* |
|---|---|---|---|---|
| **Uji 1** | *CPU Usage History* | 5%-12% | 71%-85% | 5%-19% |
| | *Networking History LAC* | 1%-5% | 20%-25% | 1%-5% |
| **Uji 2** | *CPU Usage History* | 5%-23% | 75%-90% | 5%-22% |
| | *Networking History LAC* | 1%-5% | 20%-30% | 1%-7% |
| **Uji 3** | *CPU Usage History* | 5%-16% | 70%-89% | 5%-25% |
| | *Networking History LAC* | 1%-5% | 20%-25% | 1%-5% |
| **Uji 4** | *CPU Usage History* | 5%-21% | 74%-96% | 5%-24% |
| | *Networking History LAC* | 1%-5% | 25%-45% | 1%-8% |
| **Uji 5** | *CPU Usage History* | 5%-18% | 70%-92% | 5%-17% |
| | *Networking History LAC* | 1%-5% | 25%-35% | 1%-5% |

Hasil dari proses 5 kali pengujian sebelum serangan jika dikalkulasikan rata-rata *performance* kinerja *CPU usage history* (5%-18%) dan *Networking LAC* (1%-5%), Setelah Serangan *CPU usage history* (72%-90%) dan *Networking LAC* (22%-32%) sedangkan pada saat ada serangan namun *server* jaringan sudah terpasang *honeypot* dan *port knocking performance* kinerja *CPU usage history* menjadi stabil mencapai rata-rata (5%-21%) dan *Networking LAC* (1%-6%).

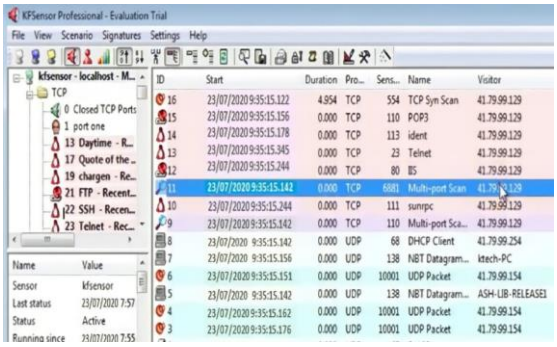
2. Mendeteksi serangan *DDoS Brute Force Attack*

Proses *exploit microtik server* dilakukan namun hasilnya gagal karena *microtik* sudah menggunakan *port knocking* untuk mengamankan server sehingga penyerang tidak bisa menembus *port-port* yang sudah ditutup dan sudah dilindungi dengan metode *port knocking* dengan status *connection error*.



Gambar 8. *Eksploit microtik* setelah pengaturan *port knocking*

Dalam proses *scanning* ditemukan *ip address* penyerang dengan metode *nmap* terdeteksi oleh *honeypot* pada *protocol TCP*.

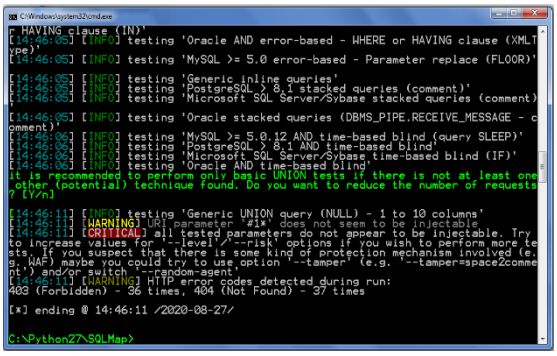


Gambar 9. *Honeypot* deteksi serangan *brute force*

Dari hasil *DDoS brute force attack* dan terdeteksi pada *honeypot* yaitu *multy port scan* dari *log* aktivitas yang masuk pada *server* jaringan.

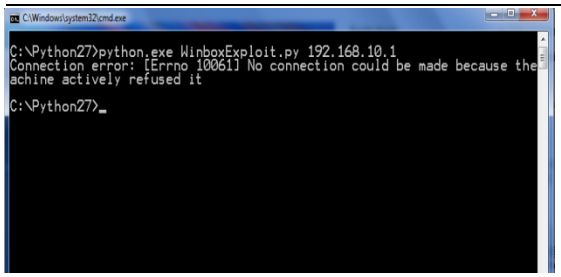
3. Mendeteksi Serangan *DDoS attack SQL injection*

Percobaan serangan ini gagal dilakukan karena *server* pada jaringan sudah terproteksi dengan baik dengan *port knocking* dan *honeypot*, seperti pada gambar 10 Serangan *SQL Injection*.



Gambar 10. Serangan *SQL Injection*

Serangan *SQL injection* tidak berhasil dapat juga dilihat pada lalulintas aktivitas pada keamanan jaringan server *honeypot* mengidentifikasi adanya *user* dari luar yang hendak mengeksploitasi pada aplikasi *computer based tes* berbasis web yang dijadikan sebagai target sasaran serangan, seperti pada gambar 11 *Honeypot* mendeteksi *DDoS SQL injection attack.*




Gambar 11. *Honeypot* mendeteksi *DDoS SQL injection attack*

Dari hasil yang terdeteksi oleh *honeypot* ada lalu lintas *user* yang hendak masuk pada jaringan ternyata penyerang akan melakukan *SQL Injection* pada *SQL server* sehingga terdeteksi pada *honeypot server*.

Tabel 8. Hasil *recovery* deteksi serangan *DDoS attack*

| **Jenis serangan** | **Deteksi serangan** | | ***Protocol*** | **Keterangan** |
|---|---|---|---|---|
| | ***Honeypot*** | ***Port Knocking*** | | |
| *DDoS attack request flooding* | *Desudes udesu-A* | *Loic* | *TCP port* 21 | Paket *request timeout and off connection* |
| *DDoS attack traffick flooding* | 192.168. 10.3 | *Hoic* | *TCP port* 21 | Paket *byte* data |
| *DDoS brute force attack* | *multy port scan* | Tidak ada log aktivitas | TCP, Port FTP 21 Port SSH 22 | *Eksploitasi username* dan *password microtik* |
| *DDoS attack SQL injection* | *SQL server* | Tidak ada log aktivitas | *TCP* | Menginjeksi *struktur database* |

Berikut hasil analisis dari semua pengujian serangan *DDoS attack*.

Tabel 9. Perbadingan hasil deteksi serangan *DDoS attack* dengan *honeypot* dan *port knocking*

| **Pengujian serangan** | **Konfigurasi honeypot dan port knocking** | | **Hasil pengujian** | |
|---|---|---|---|---|
| | **Sebelum** | **Sesudah** | **Honeypot** | **Port knocking** |
| *DDoS Attack Request Flooding* | Berhasil | Gagal | Terdeteksi | Terdeteksi |
| *DDoS Attack Traffic Flooding* | Berhasil | Gagal | Terdeteksi | Terdeteksi |
| *DDoS Brute Force Attack* | Berhasil | Gagal | Terdeteksi | - |
| *DDoS Attack SQL Injection* | Berhasil | Gagal | Terdeteksi | - |

Didapatkan presentasi dari hasil deteksi dari masing-masing serangan dan hasil deteksi dari *honeypot* dan *port knocking*, dengan menggunakan rumus:

Ph = (Jt/Jp)x100%

Ph = Presentasi hasil

Jt = Jumlah terdeteksi

Jp = Jumlah pengujian

Dengan begitu didapatkan hasil presentasi deteksi dari *honeypot* dan *port knocking*, yaitu pada *honeypot* 100% mendeteksi adanya serangan pada 4 uji coba serangan *DDoS attack* dan masing-masing serangan terdeteksi pada server *honeypot*, sedangkan pada *port knocking* 50% dari hasil pengujian 4 serangan *DDoS attack*, mendeteksi 2 serangan pada log aktivitas *microtik* dan 2 serangan lagi tidak terdeteksi karena pada percobaan awal sudah gagal dilakukan.

## 4. KESIMPULAN

Dari hasil penelitian yang sudah dilakukan dapat disimpulkan bahwa selama proses pengujian penelitian implementasi *honeypot* dan *port knocking* pada keamanan *server* jaringan dalam mendeteksi serangan dari beberapa *DDoS attack* pada *server* jaringan, sebagai berikut;

1. Pengujian serangan pada *server* jaringan *localhost* ujian jamin *offline* dari serangan *DDoS attack request flooding* dan *DDoS attack traffick flooding* berhasil terdeteksi oleh *honeypot* berupa paket *request timeout and off connection* dan paket berupa *byte* data, serangan tersebut telah dialihkan dan masuk pada *server honeypot* sehingga tidak mengganggu *server* utama jaringan dan serangan berhasil diblokir menggunakan metode *port knocking*.
2. Pengujian pada serangan *DDoS brute force attack* yaitu mencoba untuk mengeksploitasi *microtik server* untuk mendapatkan *username* dan *password* gagal dilakukan, serangan ini tidak bisa membuka *port* yang sudah dikonfigurasi dengan metode *port knocking* sehingga proses *eksploitasi* tidak berhasil dan *honeypo*t mendeteksi adanya *log* aktivitas pada *server* jaringan.

3. Pengujian serangan *DDoS attack SQL injection* dimana target serangan yaitu aplikasi web *computer based tes (CBT)* untuk mendapatkan *usename* dan *password login* sebagai *admin* gagal dilakukan karena *virtual host* pada *port* yang menjadi target serangan telah dialihkan dengan *port knocking* dan serangan ini pun dapat terdeteksi pada *server honeypot*.

## 5. SARAN

Saran untuk pengembangan selanjutnya dalam penelitian ini yaitu dalam melakukan pengujian serangan dengan menggunakan metode *DDoS attack* atau menggunakan teknik penyusupan serangan lainya sebaiknya target pengujian dalam serangan lebih banyak lagi baik secara *offline* maupun *online* agar dapat menjadi referensi dalam penelitian lanjutan dan atau lakukan pengujian serangan pada lebih dari satu *server* jaringan.

## DAFTAR PUSTAKA

[1] R. Y. Fakariah Hani Mohd Ali Mohd Azuan Mohamad Alias, "Simple Port Knocking Method," pp. 247–252, 2017.

[2] L. Catuogno, A. Castiglione, and F. Palmieri, "A honeypot system with honeyword-driven fake interactive sessions," *Proc. 2015 Int. Conf. High Perform. Comput. Simulation, HPCS 2015*, pp. 187–194, 2015, doi: 10.1109/HPCSim.2015.7237039.

[3] W. Wilman, I. Fitri, and N. D. Nathasia, "Port Knocking Dan Honeypot Sebagai Keamanan Jaringan Pada Server Ubuntu Virtual," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 3, no. 1, pp. 27–33, 2018, doi: 10.37438/jimp.v3i1.86.

[4] B. Mardiyanto, T. Indriyani, I. M. Suartana, and K. Kunci, "Analisis Dan Implementasi Honeypot Dalam Mendeteksi Serangan Distributed Denial-Of-Services (DDOS) Pada Jaringan Wireless," *Integer J.*, vol. 1, no. 2, pp. 32–42, 2016.

[5] N. Arkaan and D. V. S. Y. Sakti, "Implementasi Low Interaction Honeypot Untuk Analisa Serangan Pada Protokol SSH," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 2, pp. 112–120, 2019, doi: 10.25077/teknosi.v5i2.2019.112-120.

[6] S. J. I. I. Devie Ryana Suchendra1, Alfian Fitra Rahman2, "Penerapan sistem pengamanan port pada layanan jaringan menggunakan port knocking," *J. Lpkia*, vol. 10, no. 2, pp. 45–50, 2017.

[7] - Syaifuddin, D. Risqiwati, and E. A. Irawan, "Realtime Pencegahan Serangan Brute Force dan DDOS Pada Ubuntu Server," *Techno.Com*, vol. 17, no. 4, pp. 347–354, 2018, doi: 10.33633/tc.v17i4.1766.

[8] S. Mahajan, A. M. Adagale, and C. Sahare, "Intrusion Detection System Using Raspberry PI Honeypot in Network Security," *Int. J. Sci. Eng. Res. IJESC*, vol. 6, no. 3, pp. 2792–2795, 2016, doi: 10.4010/2016.651.

[9] M. S. Zemene and P. S. Avadhani, "Implementing high interaction honeypot to study SSH attacks," *2015 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2015*, pp. 1898–1903, 2015, doi: 10.1109/ICACCI.2015.7275895.

[10] Arbor Networks, "Worldwide Infrastructure Security Report," vol. IX, pp. 1–83, 2014, [Online]. Available: http://pages.arbornetworks.com/rs/arbor/images/WISR2014.pdf.

[11] T. Zhao, D. C. T. Lo, and K. Qian, "A neural-network based DDoS detection system using hadoop and HBase," *Proc. - 2015 IEEE 17th Int. Conf. High Perform. Comput. Commun. 2015 IEEE 7th Int. Symp. Cybersp. Saf. Secur. 2015 IEEE 12th Int. Conf. Embed. Softw. Syst. H*, pp. 1326–1331, 2015, doi: 10.1109/HPCC-CSS-ICESS.2015.38.

[12] E. Balkanli, J. Alves, and A. N. Zincir-Heywood, "Supervised learning to detect DDoS attacks," *IEEE SSCI 2014 2014 IEEE Symp. Ser. Comput. Intell. - CICS 2014 2014 IEEE Symp. Comput. Intell. Cyber Secur. Proc.*, 2014, doi: 10.1109/CICYBS.2014.7013367.