PERANCANGAN KEAMANAN SISTEM MENGGUNAKAN ALGORITMA *HONEYPOT* PADA APLIKASI KRS *ONLINE* (STUDI KASUS : SEKOLAH TINGGI TEKNOLOGI ADISUTJIPTO)

Yoris Setyawan, Haruno Sajati, Bambang Sudibya
Teknik Informatika STTA Yogyakarta
Informatika@stta.ac.id

ABSTRACT

Many colleges already apply web based academic information system use internet connection, so it is need a system security to protect from attacks such as SQL Injection and Brute Force. Honeypot is a system security which devide two system that are: original and fake system. Original system used by user to access academic information system that source from correct database. While fake system used to separate hacker and user. Hackers who try to enter with SQL Injection is send to fake system directly which has wrong data. This fake system seems correct and all action that was happened in will be noted on log history. When hacker try some password ese Brute Force, system will reset user password automatically. This system already test with 79 times SQL Injection and 29 times Brute Force attack. All attacks can handle nicely with honeypot algorithm.

Keywords: SQL injection, brute force, honeypot, system security

1. Latar Belakang Masalah

Sistem informasi akademik KRS (Kartu Rencana Studi) merupakan salah satu kemudahan bagi seorang mahasiswa untuk menginputkan matakuliah yang akan diambil pada setiap semester. Tetapi dibalik kemudahan sistem yang canggih itu perlu. Dengan sistem keamanan menggunakan *honeypot* diharapkan dapat melindungi sistem yang sedang berjalan.

Dengan adanya sistem keamanan *honeypot* diharapkan dapat melindungi sistem informasi akademik (KRS) secara maksimal dan akurat terutama dari serangan-serangan seperti *SQL injection* ataupun seperti serangan *Brute force*.

2. LANDASAN TEORI

Tinjauan Pustaka

a. *Banking Security using Honeypot*

Dalam perancangan dan pembuatan sistem keamanan pada KRS *online* ini terdapat penelitian pada jurnal karya Sandeep Chaware dari D.J.Sanghvi College of Engineering, Mumbai yang membahas tentang *banking security using honeypot*. Implementasi *honeypot* digunakan untuk memonitor dan mendeteksi menggunakan *Network Intrusion Detection System* (NIDS) atau *Intrusion Detection Systems* (IDS). Pada penelitian ini serangan hanya dipusatkan untuk *SQL injection* dan *brute force* yang diarahkan ke aplikasi KRS *online*. Sedangkan untuk monitoring tindakan *hacker* semua telah tercatat pada *log history* yang telah disediakan.

b. *Honeyweb: a web-based high interaction client honeypot*

Terdapat penelitian pada jurnal karya Nageshri B Karhade dari Vidyavardhini's College of Engineering and Technology, Mumbai University yang membahas *Honeyweb: a web-based high interaction client honeypot*. Pada penelitian ini serangan difokuskan pada tipe serangan *SQL injection* yang nantinya diarahkan ke *server* palsu oleh *honeypot*. Sedangkan untuk tipe serangan *brute force* sistem akan mengganti *password user* dengan yang baru setelah 5 kali gagal *login*.

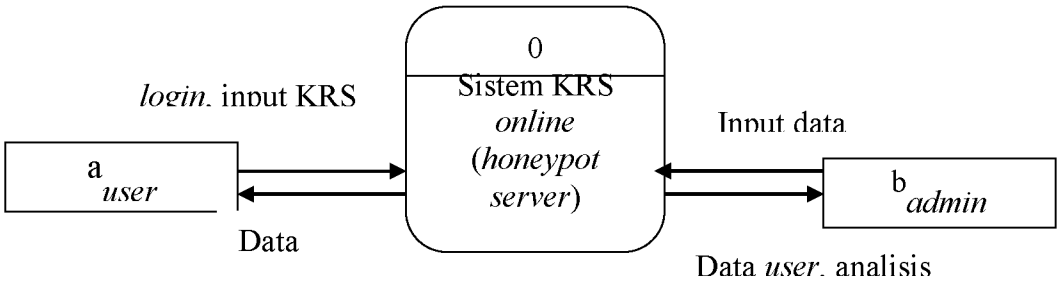
Honeypot Security

Honeypot merupakan sebuah sistem atau komputer yang sengaja "dikorbankan" untuk menjadi target serangan dari *hacker*. Komputer tersebut melayani setiap serangan yang dilakukan oleh *hacker*. Metode ini ditujukan agar administrator dari *server* yang akan diserang dapat mengetahui trik penetrasi yang dilakukan *hacker* serta bisa melakukan antisipasi dalam melindungi *server* yang sesungguhnya.

7. PERANCANGAN SISTEM

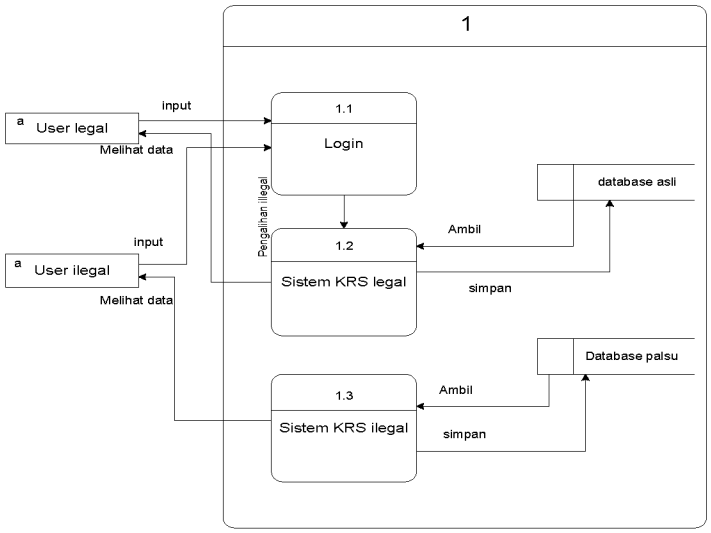
Perancangan Perangkat lunak

a. Diagram Konteks



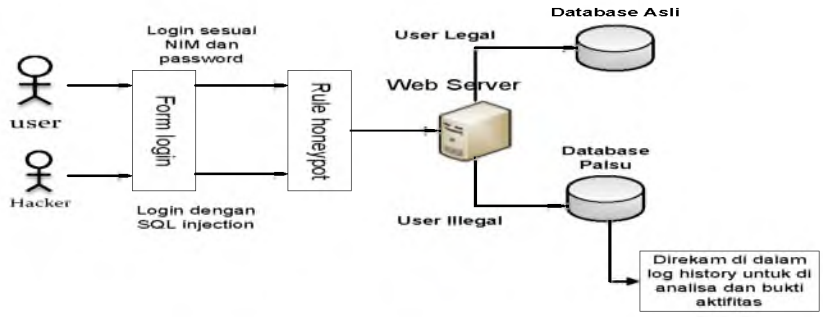
Gambar 1 Diagram Konteks

b. DFD Level 1 Aplikasi KRS *Online*



Gambar 2 DFD Level 1

Algoritma *Honeypot*



Gambar 3 Algoritma *Honeypot*

Pada Gambar 3 alur algoritma *honeypot* dapat dilihat saat *user login* dengan nim dan *password* sesuai dengan *database* maka *rule honeypot* akan memeriksa *login user* yang masuk benar sesuai dengan di *database* atau tidak. Jika benar maka *honeypot* akan menganggap *user login* dengan *legal* sehingga dapat diarahkan menuju ke sistem asli. Sedangkan saat *hacker* yang *login* dengan *SQL injection* maka *honeypot* akan memeriksa nim dan *password* yang digunakan untuk *login,* maka *hacker* akan langsung diarahkan ke sistem palsu. Selanjutnya segala aktifitas di dalam sistem palsu ke dalam *log history,* nantinya menjadi jejak yang ditinggalkan oleh *hacker* sehingga admin dapat menganalisa apa yang terjadi di dalam sistem palsu tersebut.

8. IMPLEMENTASI

User Legal

Gambar 4 Login *User*

Form *login* digunakan untuk *user* masuk ke dalam sistem KRS *online* dengan memasukkan nim dan *password* yang telah didaftarkan di form registrasi sebelumnya. Form *login* memiliki satu tombol *submit* dan dua link, yaitu registrasi dan *reset password*. Tombol *submit* digunakan untuk mengecek nim dan *password* yang telah diisi dengan data di dalam *database*

Gambar 5 Input Matakuliah

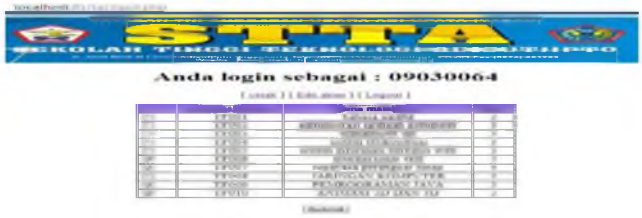
Form input matakuliah merupakan sistem utama yang dilindungi oleh *honeypot*, form input berisikan daftar-daftar matakuliah yang akan diambil oleh *user*.

Pengalihan *User Illegal*

Sistem yang digunakan untuk mengecoh *hacker* yang akan masuk ke dalam sistem dengan cara *login* menggunakan *SQL Injection*. Sistem palsu ini menggunakan *database mysql* sebagai penyimpanan data palsu yang merupakan pencerminan dari *database* asli.

Gambar 6 Pengalihan *User Illegal*

Form *login* terdapat pada Gambar 6 menunjukkan proses *login* yang dilakukan *hacker* dengan menggunakan serangan *SQL injection*. Konsep *SQL injection* menyisipkan perintah *SQL* kepada suatu statement *SQL* yang ada pada aplikasi yang sedang berjalan.



Gambar 7 Tampilan Pengalihan *User Illegal*

Tampilan pada sistem palsu dibuat sama persis dengan tampilan di sistem asli untuk mengecoh *hacker* yang menyerang dengan *SQL injection* bahkan alamat *web* dibuat sama persis dengan yang asli. Semua aktifitas *hacker* di dalam sistem palsu akan terekam dan tersimpan dalam *log history*

Pengujian

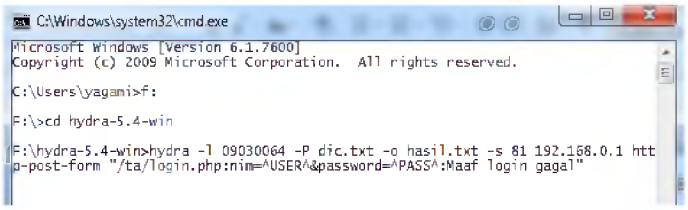
a. Uji Coba *SQL Injection*

Gambar 8 *Login Dengan SQL injection*

Proses *login* yang telah disisipkan perintah *SQL injection* pada nim *user* dengan menambahkan *command* " '-- ". *Command* ini berfungsi mengabaikan *command-command* setelah nim yang ada di dalam *script web*. Sehingga hanya dengan mengisikan nim 09030064'-- maka *hacker* akan dapat masuk ke dalam sistem, disini peran *honeypot* akan bekerja. *Honeypot* akan menyaring setiap *user* yang *login* ke dalam sistem jika nim dan *password* yang digunakan benar maka *user* akan diarahkan ke dalam *server* sedangkan jika *hacker login* dengan *SQL injection* maka *honeypot* akan mengarahkan *hacker* langsung ke sistem palsu untuk mengecoh *hacker* yang masuk. Untuk *password* dapat diisi bebas karena tidak berpengaruh terhadap nim yang telah di-*injection*.

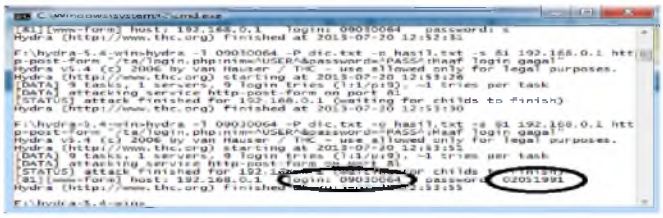
Gambar 9 Tampilan Setelah di *Hack*

b. Uji Coba *Brute Force*



Gambar 10 Mulai *Brute Force*

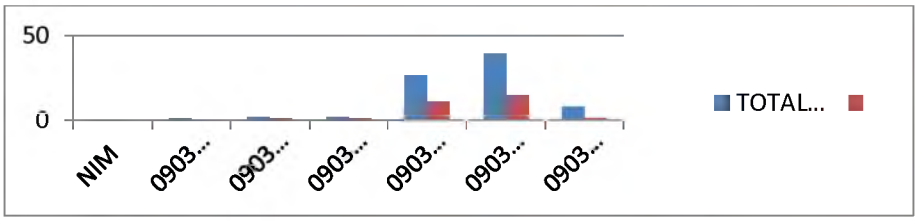
Ini merupakan tahap untuk menggunakan *software hydra* yang digunakan untuk *brute force*. Untuk menjalankan *software* ini menggunakan *command prom* atau CMD sesuaikan *directory* tempat dimana *software hydra* disimpan. *Command* untuk menggunakan *software* ini, yaitu: G:\hydra-5.4-win>hydra -l 09030064 -P dic.txt -o hasil.txt -s 81 192.168.0.1 http post form "/ta/*login*.php:nim=^*USER*^&*password*=^PASS^:Maaf *login* gagal".



Gambar 11 Hasil *Brute Force*

9. ANALISA

Dari serangan yang terjadi diperoleh data statistik berupa total serangan yang terjadi dan detail serangan berupa tanggal dan jam serangan selama pengujian.



Gambar 12 Grafik Total Serangan

Dari Gambar 12 diketahui total serangan telah terjadi 79 kali *SQL injection* dan 29 kali serangan *brute force*. Dengan pembagian, nim 09030020 diserang menggunakan *SQl injection* sebanyak 2 kali dan 1 kali dengan *brute force*, nim 09030044 diserang menggunakan *SQl injection* sebanyak 2 kali dan 1 kali dengan *brute force*, nim 09030076 diserang menggunakan *SQl injection* sebanyak 8 kali dan 1 kali dengan *brute force*, nim 09030006 diserang menggunakan *SQl injection* sebanyak 1 kali, nim 09030064 diserang menggunakan *SQl*

injection sebanyak 39 kali dan 15 kali dengan *brute force*, nim 09030071 diserang menggunakan *SQl injection* sebanyak 27 kali dan 11 kali dengan *brute force*.

10. KESIMPULAN DAN SARAN

Kesimpulan

1. Penerapan algoritma *honeypot* pada sistem KRS dapat digunakan sebagai tindakan pencegahan untuk upaya gangguan keamanan dengan *SQL injection* dan *brute force*.
2. Algoritma *honeypot* berhasil membedakan antara *user legal* dan user *illegal*.
3. Serangan dari pengguna *illegal* yang sama sebanyak lima kali akan ditindak dengan cara membuat *password* baru secara otomatis dan *random* khusus serangan *brute force*.

Saran

Kelemahan dari sistem ini yaitu *honeypot* yang dirancang tidak dapat menahan serangan berupa *virus* atau beberapa serangan yang langsung ditujukan ke *operating system* seperti *denial of service* dan *ping of death*. Saran untuk mengatasi hal tersebut dengan membangun keamanan *honeypot* dengan fungsi *firewall*. Sehingga *honeypot* dapat membedakan serangan dan dialihkan ke *server* palsu.

DAFTAR PUSTAKA

Chaware, Sandeep., 2011., *Banking Security Using Honeypot,* International Journal of Security and Its Applications Vol. 5 No. 1 hal 31, D.J.Sanghvi College of Engineering, Mumbai, India.

Jogiyanto, MBA, Akt., *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori Dan Praktik Aplikasi Bisnis,* Penerbit Andi, Yogyakarta.

Patil, Sainath, dkk., 2012., *Honeyweb: a web-based high interaction client honeypot,* International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vidyavardhini's College Of Engineering And Technology, Mumbai, India.

Utdirartatmo, Firrar., 2005., *Menjebak Hacker dengan Honeypot,* Penerbit Andi offset, Yogyakarta.