**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : Abdurrazak Baihaqi**

**NRP : 5110 100 027**

**DOSEN WALI : Dr.Eng. Nanik Suciati, S.Kom, M.Kom**

**DOSEN PEMBIMBING : 1. Ary Mazharuddin Shiddiqi, S.Kom, M.Comp.Sc  
 2. Baskoro Adi Pratomo, S.Kom, M.Kom**

# JUDUL TUGAS AKHIR

“**Perancangan *Web Application Honeypot* untuk menggali informasi mengenai peretas**”

# LATAR BELAKANG

Aplikasi *web* adalah sebuah aplikasi yang berjalan diatas sebuah *web browser* yang dirancang menggunakan bahasa pemrograman yang didukung oleh *web browser*. Aplikasi *web* dibangun dalam arsitektur *client-server*, yang memanfaatkan *web browser* sebagai *thin client* [1]. Kemudahan dalam pengoperasian aplikasi *web*, beserta fitur *cross-platform* membuat penggunaan aplikasi *web* menjadi lebih populer dibanding penggunaan aplikasi *desktop* yang berbasis *client-server*. Perkembangan internet dan *World Wide Web* (WWW) yang pesat turut membuat penggunaan aplikasi *web* menjadi semakin populer.

Pertumbuhan internet yang pesat dan penggunaan aplikasi *web* yang semakin populer menyebabkan aplikasi *web* seringkali menjadi target utama serangan para peretas. Berbagai kelemahan dari aspek keamanan mendudukkan aplikasi *web* pada peringkat pertama target serangan *cross-platform* berdasarkan survei yang dilakukan oleh OWASP [2]. Beberapa serangan yang umum pada aplikasi *web* antara lain *Cross Site Scripting* (XSS) dan SQL *Injection*, yang dilakukan dengan memanfaatkan kelemahan aplikasi web yang tidak melakukan validasi dan sanitasi pada input data. Selain itu terdapat pula jenis serangan lain, seperti *Code Execution*, *Remote File Inclusion*, *Local File Inclusion*, dan beberapa serangan lainnya. Serangan-serangan terhadap aplikasi *web* yang dilakukan oleh peretas membuat beberapa pihak menginisiasi untuk membuat sebuah proyek yang bertujuan membuat sebuah sistem yang dirancang khusus untuk mengamati perilaku dan serangan yang dilakukan oleh peretas. Proyek ini disebut dengan *Honeypot Project*.

*Honeypot* adalah sebuah sistem yang dibuat untuk mensimulasikan layanan yang berjalan diatas sebuah server, dengan tujuan sebagai umpan untuk mengamati pola serangan yang dilakukan oleh peretas. *Honeypot* dibagi menjadi dua jenis, yakni *High-Interaction Honeypot* dan *Low-Interaction Honeypot*. *High-Interaction Honeypot* adalah *honeypot* yang mensimulasikan berbagai *vulnerability* yang terdapat diatas sebuah *server*, contohnya adalah *High Interaction Honeypot Analysis Toolkit* (HIHAT) [3]. *Low Interaction Honeypot* adalah *honeypot* yang mensimulasikan *vulnerability* tertentu yang sering menjadi target serangan para peretas, contohnya adalah *Glastopf*[4]. Melalui data yang dikumpulkan oleh *honeypot*, kita bisa mempelajari tentang pola-pola serangan yang dilakukan oleh peretas. Dengan mempelajari pola-pola serangan, maka bisa dirumuskan hal-hal yang perlu dilakukan untuk melindungi *production system* dari ancaman peretas. Karena *honeypot* dirancang sebagai umpan untuk menarik perhatian peretas, *honeypot* juga memiliki potensi untuk memberikan reaksi terhadap serangan yang dilakukan oleh peretas, salah satunya adalah untuk mengumpulkan informasi-informasi tentang identitas peretas.

Pada tugas akhir ini akan dibangun sebuah *web application* *honeypot* yang dapat mensimulasikan beberapa *vulnerability* yang terdapat pada aplikasi web sekaligus menggali informasi mengenai peretas dengan memanfaatkan kode Javascript. *Honeypot* akan mengidentifikasi *request* yang dikirim oleh *browser*. Jika *request* itu adalah *request* HTTP yang normal, maka honeypot akan memberikan *response* yang normal pula. Namun apabila pada request tersebut terdapat indikasi serangan, maka honeypot akan menyisipkan sebuah kode Javascript ke dalam *response* yang akan dikirim balik kepada *browser*. Kode Javascript itu berfungsi untuk mengumpulkan informasi-informasi tertentu, yang akan dieksekusi dengan memanfaatkan browser peretas. Setelah browser peretas mengeksekusi kode Javascript tersebut, informasi yang diperoleh akan dikirim kembali menuju sistem *honeypot* untuk dikumpulkan. Dengan adanya sistem ini, diharapkan *honeypot* pada masa yang akan datang memiliki nilai kegunaan yang lebih, karena selain dapat digunakan untuk mengumpulkan informasi mengenai pola-pola penyerangan yang dilakukan oleh peretas, *honeypot* juga mampu digunakan untuk menggali informasi mengenai peretas.

# RUMUSAN MASALAH

Berikut beberapa hal yang menjadi rumusan masalah dalam tugas akhir ini:

1. Bagaimana mensimulasikan *vulnerability* yang terdapat pada aplikasi *web* pada *honeypot*?
2. Bagaimana mendeteksi adanya indikasi serangan pada *request* HTTP yang dikirim oleh *client*?
3. Bagaimana cara mengumpulkan informasi yang berkaitan dengan identitas penyerang dengan memanfaatkan Javascript?

# BATASAN MASALAH

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada tugas akhir ini, yaitu:

1. Sistem *Honeypot* yang dibangun bersifat *low-interactive*
2. Serangan yang dapat dideteksi dan disimulasikan pada system *honeypot* ini adalah SQL *Injection* dan XSS (*Cross-site Scripting*).
3. Kode Javascript yang disisipkan pada *response* HTTP hanya dapat dieksekusi melalui *browser*
4. Tampilan aplikasi *web* yang disimulasikan melalui sistem *honeypot* ini bersifat statis

# TUJUAN PEMBUATAN TUGAS AKHIR

Tugas akhir dibuat dengan beberapa tujuan. Berikut beberapa tujuan dari pembuatan tugas akhir:

1. Mampu mensimulasikan *vulnerability* yang berupa SQL *Injection* dan XSS (*Cross Site Scripting*) pada *honeypot* yang dibangun
2. Mampu menggali informasi tentang peretas yang mengakses *honeypot* yang dibangun dengan memanfaatkan kode Javascript

# MANFAAT TUGAS AKHIR

Dengan dibangunnya sistem *honeypot* ini, diharapkan mampu menunjukkan pada pengguna bahwa *honeypot* dapat dimanfaatkan sebagai perangkap untuk mengumpulkan informasi mengenai identitas peretas.

# TINJAUAN PUSTAKA

* 1. *Honeypot*

*Honeypot* adalah sebuah server atau sistem yang digunakan sebagai umpan untuk mengumpulkan informasi tentang penyerang atau penyusup yang masuk ke dalam sistem kita. *Honeypot* dapat dipasang di dalam atau diluar *firewall* DMZ (*Demilitarized Zone*), meskipun pada umumnya *honeypot* dipasang di dalam DMZ untuk kemudahan akses dan kontrol. *Honeypot* juga bisa disebut sebagai varian dari sebuah IDS (*Intrusion Detection System*), yang fungsinya lebih difokuskan untuk pengumpulan informasi dan umpan untuk menipu peretas. [5]

Terdapat beberapa jenis *honeypot* menurut jenis layanan yang disimulasikan. Salah satunya adalah *web application honeypot*. Web application honeypot adalah honeypot yang mensimulasikan *vulnerability* pada aplikasi web sebagai umpan untuk menarik perhatian peretas. *Web application honeypot* dibagi menjadi dua jenis berdasarkan cara berinteraksi dengan klien, yakni *high interaction* dan *low interaction*. Pada tugas akhir ini, yang akan dibangun adalah *web application honeypot* yang bersifat *low interaction*.

* 1. Javascript

Javascript adalah bahasa pemrograman komputer yang dinamis. Javascript umumnya digunakan sebagai bagian dari *web browser*, yang implementasinya memungkinkan untuk dieksekusi secara *client-side*, mengontrol *browser*, berkomunikasi secara asinkronus, dan mengubah isi dokumen yang ditampilkan oleh *web browser*.[6]

Karena javascript bersifat *client side*, javascript dapat dimanfaatkan untuk menggali informasi mengenai klien, ketika kode javascript dieksekusi pada *web browser* klien. Beberapa informasi yang bisa didapatkan melalui *web browser* klien dengan memanfaatkan javascript antara lain : informasi mengenai sistem operasi dan *web browser* yang digunakan klien, mengetahui posisi klien secara geografis menggunakan informasi alamat IP, mendeteksi apakah klien sedang *login* pada *website* tertentu, dan mengungkap identitas klien melalui *clickjacking*.[7] Pada tugas akhir ini, javascript akan disisipkan pada *response* yang dikirim oleh *honeypot*.

# RINGKASAN ISI TUGAS AKHIR

Pertumbuhan internet yang pesat dan penggunaan aplikasi *web* yang semakin populer menyebabkan aplikasi *web* seringkali menjadi target utama serangan para peretas. Untuk mengamati perilaku dan pola serangan yang dilakukan oleh peretas, diperlukan adanya sistem yang digunakan sebagai umpan untuk menarik perhatian peretas agar melakukan serangan terhadap sistem tersebut. Sistem ini disebut dengan *web application honeypot*.

Tujuan dari pembuatan tugas akhir ini adalah untuk merancang sebuah *web application honeypot* yang mensimulasikan beberapa *vulnerability* pada aplikasi *web*, yang dapat dieksploitasi dengan serangan SQL *Injection* dan *Cross Site Scripting*. Apabila *honeypot* menerima *request* HTTP yang terindikasi sebagai sebuah serangan, *honeypot* akan mengirimkan *response* kepada klien sesuai dengan klasifikasi serangan yang dilakukan oleh klien Berikut akan dijelaskan

Langkah – langkah pengerjaan tugas akhir ini akan dijelaskan pada Gambar 1.



Gambar 1. Alur pengerjaan proposal tugas akhir

Berikut alur kerja dari aplikasi:



Gambar 2. Alur kerja aplikasi

Berdasarkan gambar 2 diatas, dapat diketahui bahwa *honeypot* mengirimkan *response* HTTP kepada klien sesuai dengan klasifikasi serangan pada *request* HTTP yang dikirim oleh klien. Ketika klien mengirim request HTTP menuju *honeypot*, *honeypot* akan memeriksa *request* tersebut. *Request* kemudian diklasifikasikan sesuai dengan indikasi serangan. *Honeypot* kemudian akan membuat pesan *response* HTTP sesuai dengan hasil klasifikasi *request*. Pada *response* tersebut kemudian disisipkan kode javascript untuk menggali informasi mengenai peretas. Kode javascript tersebut akan dieksekusi ketika *response* yang dikirim oleh *honeypot* ditampilkan pada *web browser*.

# METODOLOGI

## Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisikan mengenai pembangunan sebuah *honeypot* yang dibuat untuk mensimulasikan *vulnerability* terhadap serangan SQL *Injection* dan *Cross Site Scripting*. Apabila *honeypot* menerima *request* yang terindikasi sebagai serangan SQL *Injection* atau *Cross Site Scripting*, maka *honeypot* akan memberikan *response* berupa tampilan yang seolah-olah menunjukkan adanya *vulnerability* dalam *honeypot*. Kemudian, informasi mengenai peretas dapat digali melalui eksekusi kode Javascript pada *web browser* peretas. Javascript tersebut disisipkan dalam *response* yang dikirim oleh *honeypot*.

## Studi literatur

Tugas akhir ini menggunakan literatur *paper* beserta artikel dari internet. *Paper* yang digunakan adalah *“Know Your Tools: Glastopf - A dynamic, low-interaction web application honeypot”*. *Paper* tersebut menjadi acuan utama dan dasar dalam pengerjaan tugas akhir ini.

## Analisis dan Desain Perangkat Lunak

Sistem *Honeypot* yang dibangun menggunakan arsitektur klien *–* server*.* *Honeypot* yang dibangun dijalankan pada sebuah server, sedangkan yang menjadi klien adalah *web browser* yang digunakan oleh peretas.Gambar 3 menunjukkan diagram arsitektur jaringan yang dibutuhkan oleh aplikasi.



Gambar 3. Arsitektur jaringan

## Implementasi perangkat lunak

Dalam pembuatan aplikasi, digunakan beberapa teknologi untuk dapat mengaplikasikan rancangan yang sudah ada, diantaranya:

1. Bahasa Pemrograman Aplikasi

Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Python. Penggunaanbahasa pemrogramandiharapkan dapat membantu menangani kebutuhan aplikasi terutama kemudahan untuk konektivitas dengan basis data dan kebutuhan lainnya.

1. Basis Data

Basis data pada serverdigunakan untuk menampung *log* dari *request* yang dikirim oleh klien, sekaligus untuk menyimpan informasi dari klien yang berhasil dihimpun melalui eksekusi kode Javascript. Dalam sistem ini akan digunakan basis data MySQL.

1. IDE

Pengembangan aplikasi ini menggunakan Vim, sebuah *text editor* sebagai IDE.

1. Modeling Tools

Beberapa *modeling tools* yang digunakan untuk mengembangkan aplikasi ini Power Designer 15.00, StarUML, Microsoft Visio 2013.

## Pengujian dan Evaluasi

Pada tahap ini akan dilakukan pengujian terhadap *honeypot* yang dibangun dengan cara mengirim *request* HTTP yang ditujukan pada *honeypot*, untuk memastikan *honeypot* yang telah dibangun mampu mengenali serangan yang berupa SQL *Injection* dan *Cross Site Scripting*, sekaligus mampu memberikan *response* yang sesuai dengan *request* yang dikirim oleh klien.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2014 | | | | | | | | | | | | | | | | | | | | |
| Maret | | | | April | | | | Mei | | | | Juni | | | | | Juli | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | “Web Application,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Web\_application . [Accessed 24 February 2014] |
| [2] | “OWASP Top 10,” OWASP, [Online]. Available: https://www.owasp.org/index.php/Top\_10\_2013-Top\_10.  [Accessed 24 February 2014]. |
| [3] | “HIHAT,” HIHAT Honeypot Project, [Online]. Available: http://hihat.sourceforge.net. [Accessed 24 February 2014]. |
| [4] | “Glastopf,” Glastopf Honeypot Project, [Online]. Available: http://glastopf.org/. [ Accessed 24 February 2014]. |
| [5] | “Honey Pot Systems Explained,” SANS Security Resources, [Online]. Available: http://www.sans.org/security-resources/idfaq/honeypot3.php/.  [ Accessed 24 February 2014]. |
| [6] | “Javascript,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/JavaScript /. [Accessed 24 February 2014]. |
| [7] | “I Know Series,” WhiteHat Security Blog, [Online]. Available: http://blog.whitehatsec.com/introducing-the-i-know-series/.  [Accessed 24 February 2014]. |
| [8] | Lukas Rist, “ Know Your Tools: Glastopf - A dynamic, low-interaction web application honeypot”. |