JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

USULAN TUGAS AKHIR

1. 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.

2. JUDUL TUGAS AKHIR

"**Perancangan *Web Application Honeypot* untuk Menggali Informasi Peretas**"

3. LATAR BELAKANG

Aplikasi *web* adalah sebuah aplikasi yang berjalan di atas 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* yang pesat turut membuat penggunaan aplikasi *web* menjadi semakin populer.

Pertumbuhan internet yang pesat dan penggunaan aplikasi *web* yang semakin populer menyebabkan aplikasi *web* sering kali 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 ditujukan pada aplikasi *web* adalah XSS (*Cross-Site Scripting*) dan SQL *injection* yang dilakukan dengan memanfaatkan

kelemahan aplikasi *web* yang tidak melakukan validasi dan sanitasi pada data masukan. 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 menyimulasikan layanan yang berjalan di atas 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 menyimulasikan berbagai *vulnerability* yang terdapat pada sebuah *server*, contohnya adalah *High Interaction Honeypot Analysis Toolkit* (HIHAT) [3]. *Low-interaction honeypot* adalah *honeypot* yang menyimulasikan *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 menyimulasikan 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 tersebut 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.

4. RUMUSAN MASALAH

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

a. Bagaimana menyimulasikan *vulnerability* yang terdapat pada aplikasi *web* pada *honeypot*?
b. Bagaimana mendeteksi adanya indikasi serangan pada *request* HTTP yang dikirim oleh *client*?
c. Bagaimana cara mengumpulkan informasi yang berkaitan dengan identitas penyerang dengan memanfaatkan JavaScript?

5. BATASAN MASALAH

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

a. *Honeypot* yang dibangun bersifat *low-interactive*.
b. Serangan yang dapat dideteksi dan disimulasikan pada *honeypot* ini adalah SQL *injection* dan *Cross-Site Scripting*.
c. Kode JavaScript yang disisipkan pada *response* HTTP hanya dapat dieksekusi melalui *browser*.
d. Tampilan aplikasi *web* yang disimulasikan melalui sistem *honeypot* ini bersifat statis.

6. TUJUAN PEMBUATAN TUGAS AKHIR

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

a. Mampu menyimulasikan *vulnerability* yang berupa SQL *injection* dan *Cross-Site Scripting* pada *honeypot* yang dibangun.
b. Mampu menggali informasi tentang peretas yang mengakses *honeypot* yang dibangun dengan memanfaatkan kode JavaScript.

7. MANFAAT TUGAS AKHIR

Dengan dibangunnya sistem *honeypot* ini diharapkan pengguna dapat mengamati pola-pola serangan yang dilakukan oleh peretas yang tercatat oleh *honeypot.* Selain itu pengguna dapat memperoleh informasi mengenai peretas yang diperoleh melalui eksekusi kode JavaScript pada *browser* yang disisipkan pada *response* HTTP yang dikirim oleh *honeypot*.

8. TINJAUAN PUSTAKA

a. 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 di luar *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 menyimulasikan *vulnerability* pada aplikasi *web* sebagai umpan untuk menarik perhatian peretas. *Web application honeypot* dibagi menjadi dua jenis berdasarkan cara berinteraksi dengan *client*, yakni *high-interaction* dan *low-interaction*. Pada tugas akhir ini akan dibangun *web application honeypot* yang bersifat *low-interaction*.

b. JavaScript

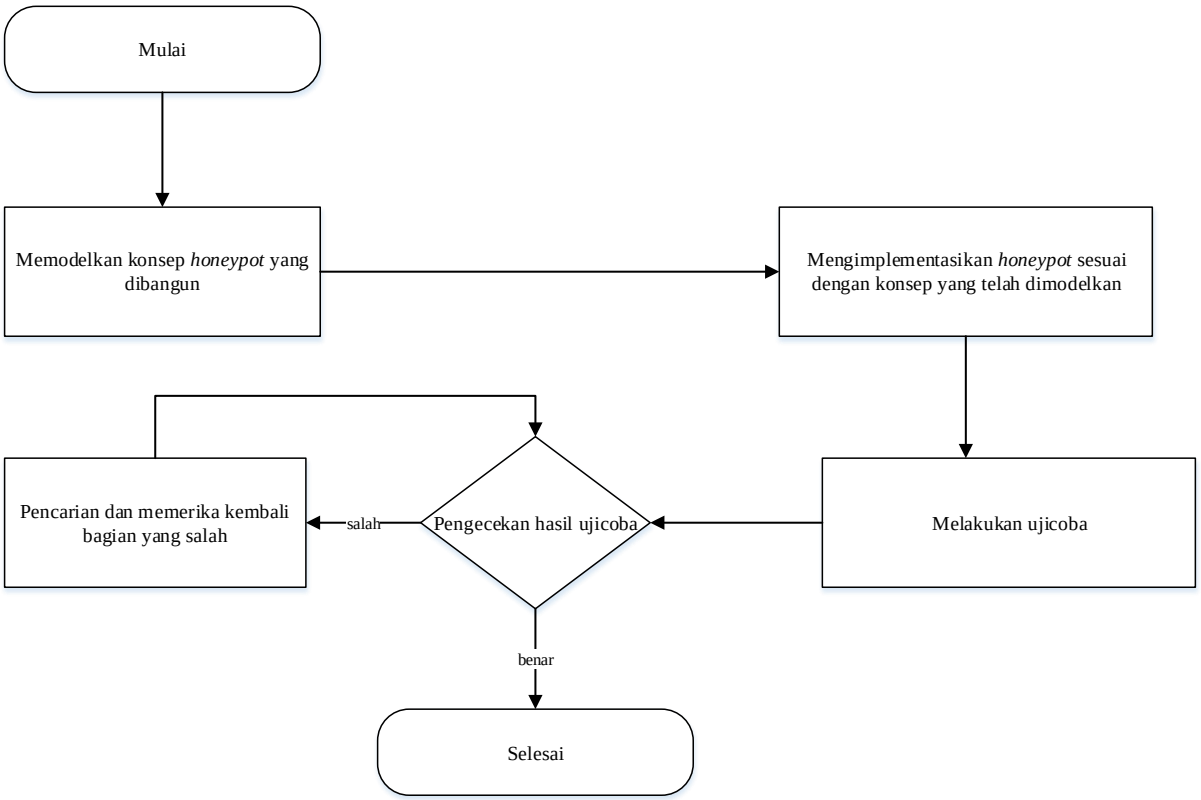
JavaScript adalah bahasa pemrograman komputer yang dinamis. JavaScript umumnya digunakan sebagai bagian dari *web browser* yang implementasinya memungkinkan untuk dieksekusi secara *client-side*, mengontrol *browser*, berkomunikasi secara asinkronus, dan mengubah isi dokumen yang ditampilkan oleh *web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

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

9. RINGKASAN ISI TUGAS AKHIR

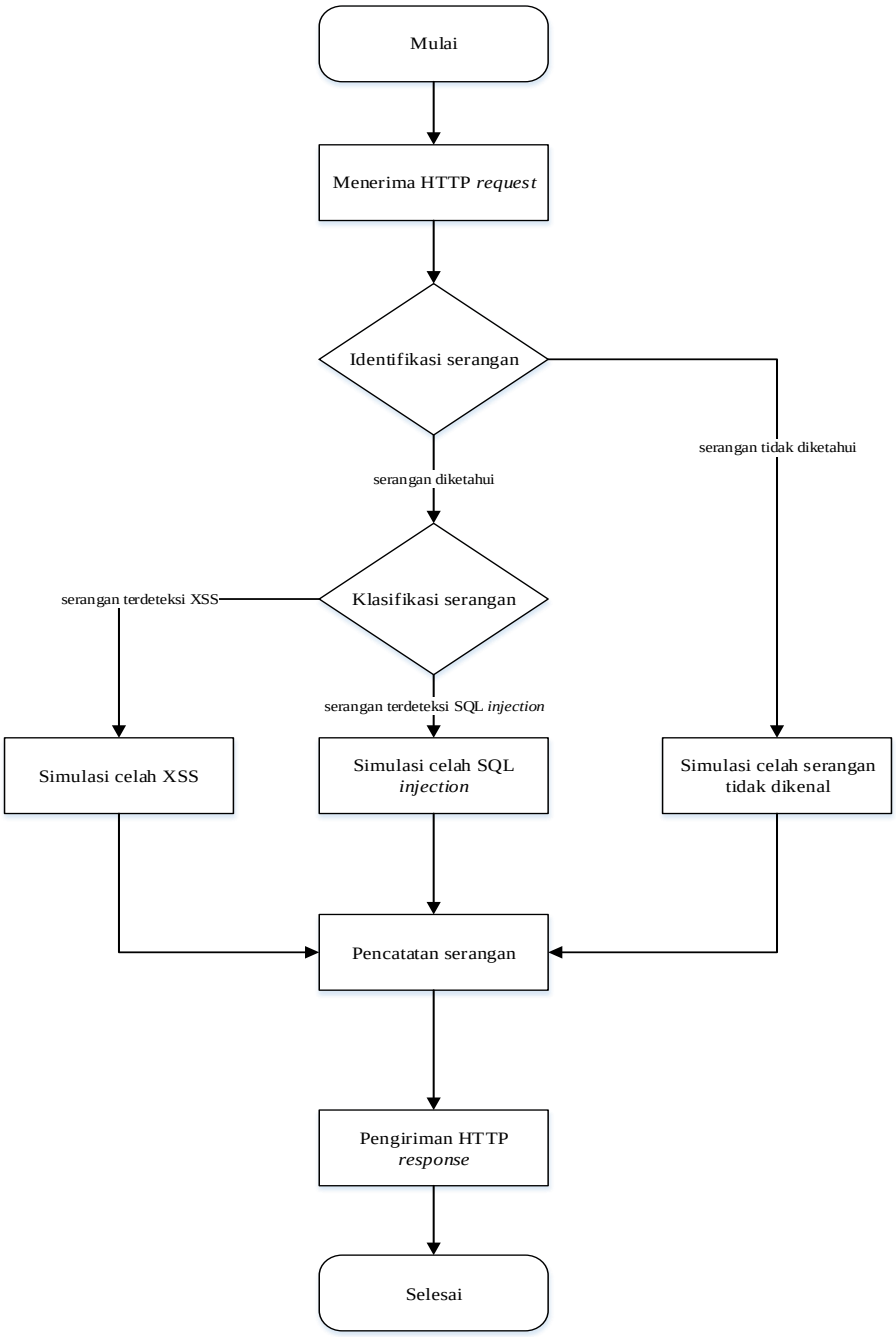
Pertumbuhan internet yang pesat dan penggunaan aplikasi *web* yang semakin populer menyebabkan aplikasi *web* seringkali menjadi target utama serangan para peretas. Untuk mengamati perilaku dan pola serangan yang dilakukan oleh peretas diperlukan adanya sistem yang digunakan sebagai umpan untuk menarik 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 menyimulasikan 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 *client* sesuai dengan klasifikasi serangan yang dilakukan oleh *client* Langkah-langkah pengerjaan tugas akhir dijelaskan sesuai pada Gambar 1.



Gambar 1. Alur pengerjaan proposal tugas akhir

Berikut alur kerja dari aplikasi:



Gambar 2. Alur kerja aplikasi

Berdasarkan Gambar 2 di atas, dapat diketahui bahwa *honeypot* mengirimkan *response* HTTP kepada *client* sesuai dengan klasifikasi serangan pada *request* HTTP yang dikirim oleh *client*. Ketika *client* mengirim *request* HTTP menuju *honeypot*, *honeypot* akan memeriksa *request* tersebut. *Request* kemudian diklasifikasi 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*.

10.METODOLOGI

a. Penyusunan proposal tugas akhir

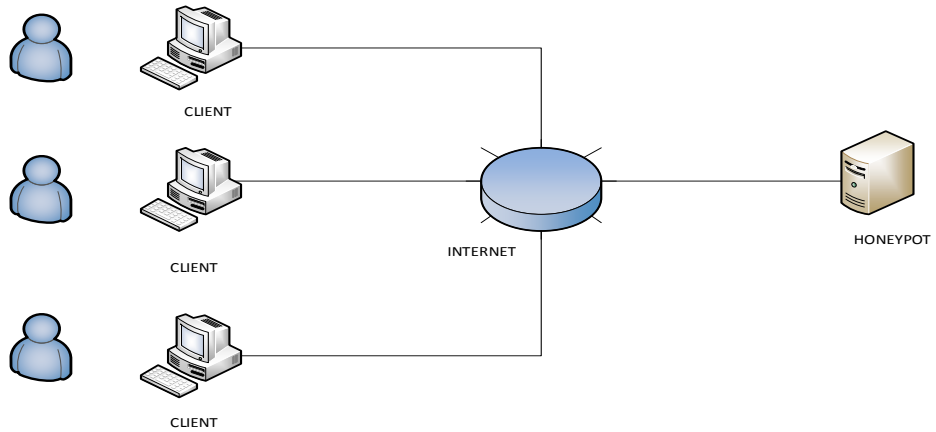
Proposal tugas akhir ini berisikan mengenai pembangunan sebuah *honeypot* yang dibuat untuk menyimulasikan *vulnerability* terhadap serangan SQL *injection* dan *Cross-Site Scripting*. Apabila *honeypot* menerima *request* yang terindikasi sebagai serangan SQL *injection* atau *Cross-Site Scripting*, 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*.

b. Studi literatur

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

c. Analisis dan Desain Perangkat Lunak

Sistem *Honeypot* yang dibangun menggunakan arsitektur *client-server*. *Honeypot* akan dijalankan pada sebuah *server* dan yang menjadi *client* adalah *web browser* yang digunakan oleh peretas. Gambar 3 menunjukkan diagram arsitektur jaringan yang dibutuhkan oleh aplikasi.



Gambar 3. Arsitektur jaringan

d. Implementasi perangkat lunak

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

a. Bahasa Pemrograman Aplikasi
   Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Python. Penggunaan bahasa pemrograman diharapkan dapat membantu menangani kebutuhan aplikasi terutama kemudahan untuk memproses *request* HTTP dan kebutuhan lainnya.
b. Basis Data
   Basis data pada *server* digunakan untuk menampung *log* dari *request* yang dikirim oleh *client*, sekaligus untuk menyimpan informasi dari *client* yang berhasil dihimpun melalui eksekusi kode JavaScript. Dalam sistem ini akan digunakan basis data MySQL.
c. IDE
   Pengembangan aplikasi ini menggunakan Vim, sebuah *text editor* sebagai IDE.
d. Modeling Tools
   Beberapa *modeling tools* yang digunakan untuk mengembangkan aplikasi ini Power Designer 15.00, StarUML dan Microsoft Visio 2013.

e. 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 berupa SQL *injection* dan *Cross-Site Scripting*, serta mampu memberikan *response* yang sesuai dengan *request* yang dikirim oleh *client*.

f. 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
    a. Latar Belakang
    b. Rumusan Masalah
    c. Batasan Tugas Akhir
    d. Tujuan
    e. Metodologi
    f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Jadwal kegiatan pengerjaan tugas akhir ini dapat dilihat pada Tabel 1. Pengerjaan tugas akhir ini dibagi menjadi beberapa tahap yaitu tahap penyusunan proposal, tahap studi literatur, tahap perancangan sistem, tahap implementasi, tahap pengujian dan evaluasi, dan tahap penyusunan buku.

Tabel 1. Jadwal Kegiatan

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

12. DAFTAR PUSTAKA

[1] "Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Web_application. [Accessed 24 February 2014].

[2] "OWASP Top Ten project," [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Accessed 24 February 2014].

[3] "HIHAT Honeypot Project," [Online]. Available: http://hihat.sourceforge.net/. [Accessed 24 February 2014].

[4] L. Rist, "Honeynet Project," [Online]. Available: http://honeynet.org/sites/default/files/files/KYT-Glastopf-Final_v1.pdf. [Accessed 24 February 2014].

[5] "SANS," [Online]. Available: http://www.sans.org/security-resources/idfaq/honeypot3.php. [Accessed 24 February 2014].

[6] "Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/JavaScript. [Accessed 24 February 2014].

[7] "Whitehat Security Blog," [Online]. Available: http://blog.whitehatsec.com/introducing-the-i-know-series/. [Accessed 24 February 2014].