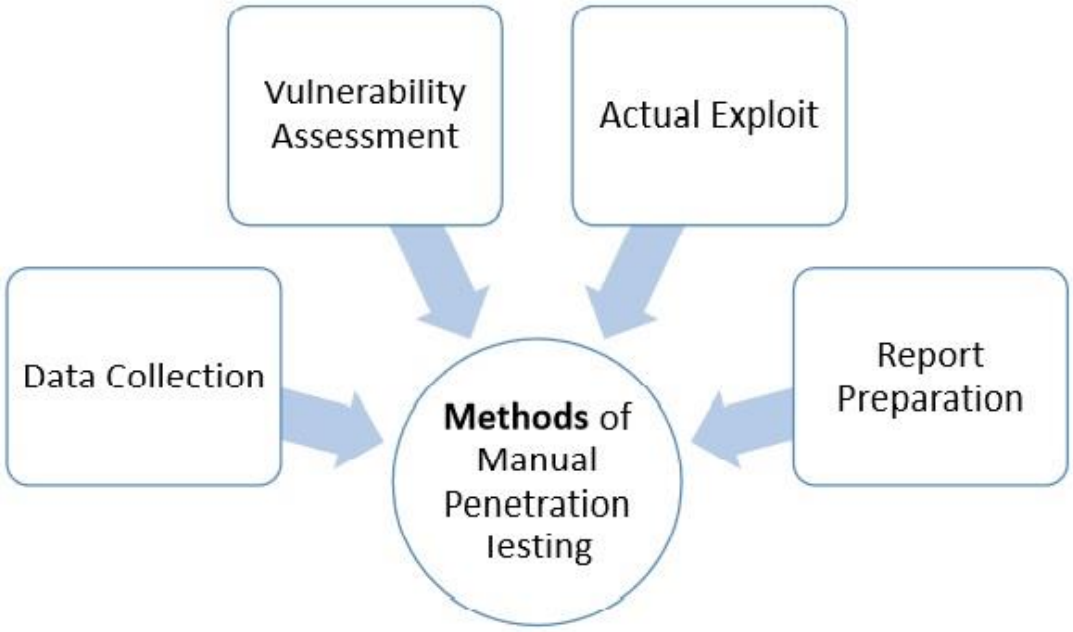
ACTUAL EXPLOIT

Exploit adalah sebuah kode yang menyerang keamanan komputer secara spesifik. Exploit banyak digunakan untuk penetrasi baik secara legal ataupun ilegal untuk mencari kelemahan (*vulnerability*) pada komputer tujuan.



Gambar 1: langkah-langkah *penetration testing step*

Untuk melakukan exploit kita harus mencari sumber lubang keamanan pada target. Lubang keamanan (*security hole*) dapat terjadi karena kesalahan konfigurasi, salah implementasi, dan salah penggunaan.

Untuk melakukan exploit dibutuhkan :

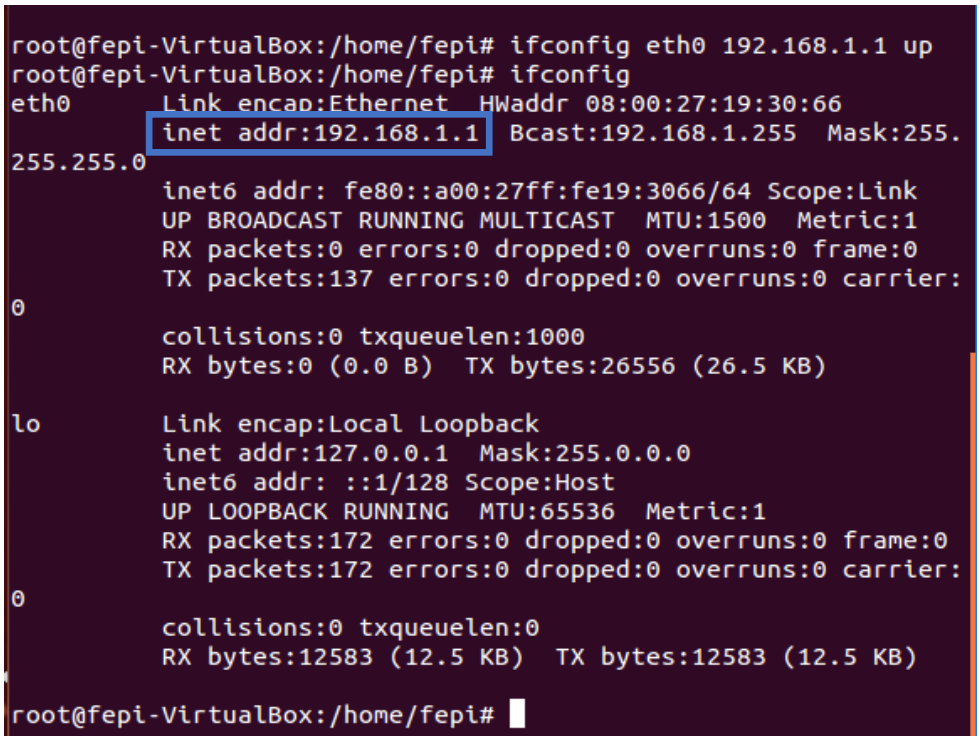
- 1 buah laptop
- Aplikasi VirtualBox/VMware
- iso Ubuntu-Dekstop 14.04 32bit
- iso DVL (*Damn Vulnerable Linux*) Linux-Kernel 2.6

Berikut adalah penjelasan secara teknis yang telah dilakukan untuk percobaan exploit:

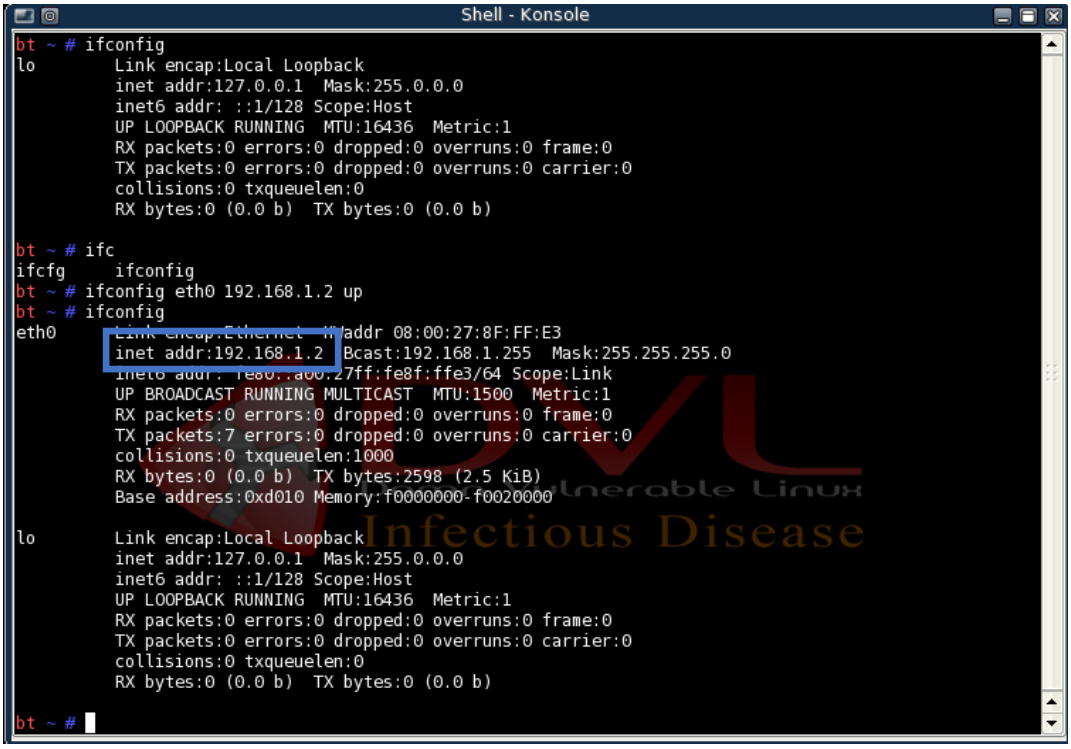
- Install VirtualBox atau VMware, yang berguna untuk menjalankan banyak mesin dalam satu laptop. Jika telah di install langsung ke tahap 2.

- Lakukan install OS iso Ubuntu-Dekstop 14.04 32bit dan iso DVL di VirtualBox atau VMware. OS DVL digunakan sebagai target dan OS Ubuntu digunakan sebagai sistem yang mencoba mencari informasi tentang target, karena di OS Ubuntu dilakukan *scanning* untuk mencari *vurnerability* dari target.

- Setelah tahap 2 selesai, lakukan konfigurasi IP address pada masing-masing mesin, selain itu *network*-nya pun harus disamakan agar Ubuntu dan DVL dapat saling terhubung. Pada percobaan ini, penulis memberi IP address Ubuntu yaitu 192.168.1.1 dan DVL 192.168.1.2.



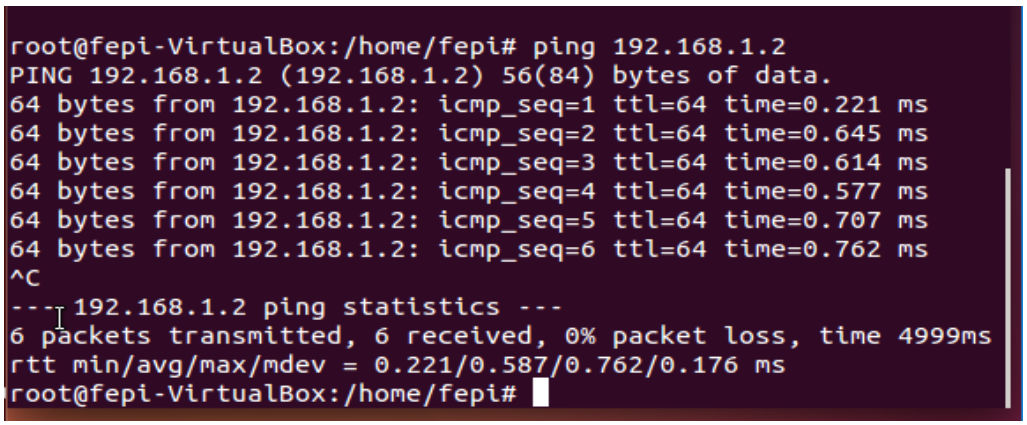
Gambar 2: set up *ip address* Ubuntu



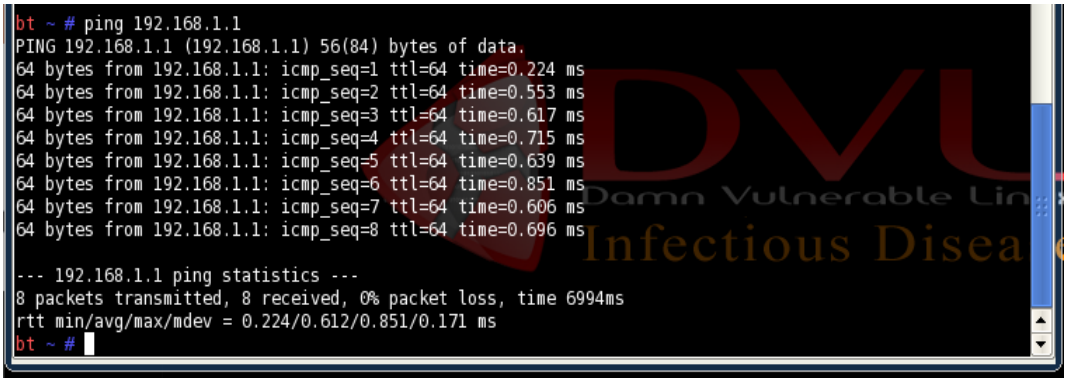
Gambar 3: set up *ip address* DVL

Command **ifconfig eth0 [ip address] up** digunakan untuk memberi ip address secara sementara, maksudnya yaitu jika mesin di *power-off* (dimatikan) maka ip address yang telah dikonfigurasi akan hilang lagi. Jika tak ingin ip address pada mesin menghilang, bisa dilakukan konfigurasi ip static melalui *command* **nano etc/network/interfaces** kemudian masukan ip address, netmask, dan gateway, lalu simpan.

- Jika langkah 3 telah dilakukan, lakukan perintah ping untuk mengecek apakah mesin Ubuntu dan DVL sudah terhubung atau tidak. Jika telah terhubung lanjut ke langkah selanjutnya.

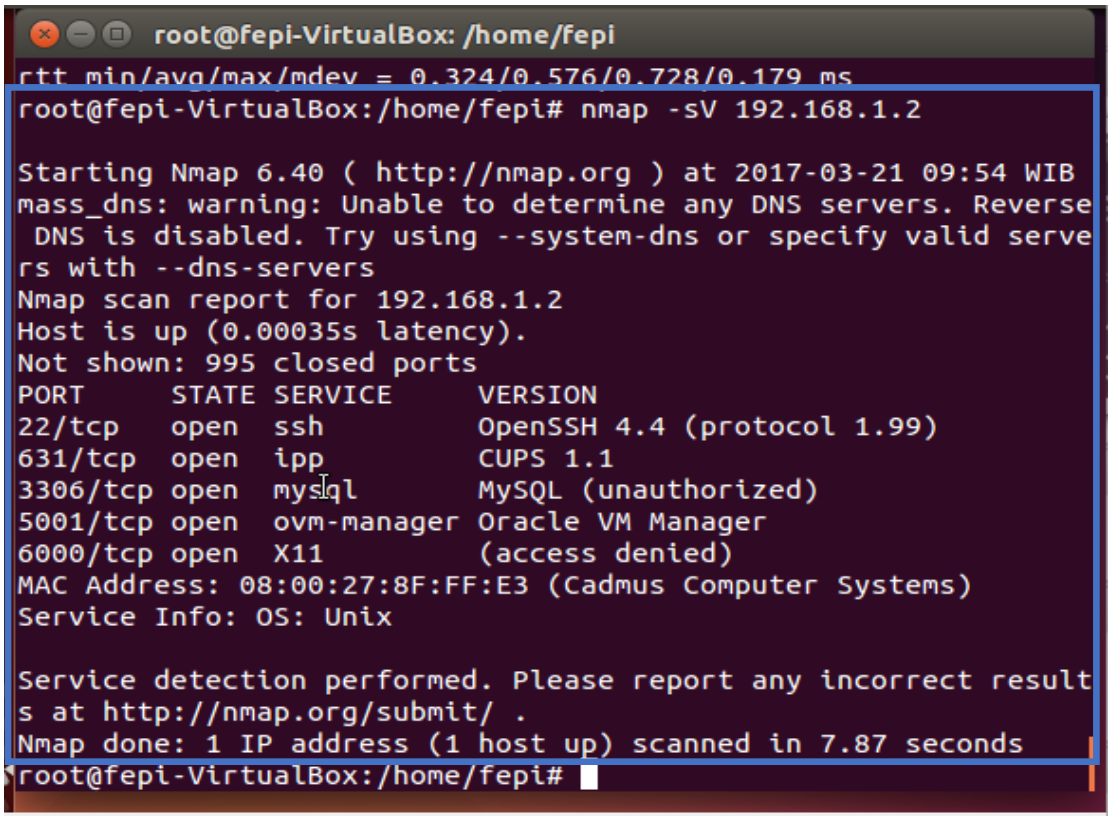


Gambar 4: ping *ip address* DVL melalui Ubuntu



Gambar 5: ping *ip address* Ubuntu melalui DVL

- Setelah Ubuntu dan DVL telah terhubung, kita lakukan *scanning* untuk melihat *service* yang terbuka pada target. Ubuntu diibaratkan sebagai pentester, sedangkan DVL diibarakat sebagai target.



Gambar 6: proses *scanning* pada Ubuntu

Berdasarkan gambar 6, dapat kita lihat ada beberapa layanan yang statusnya *Open* pada target. Untuk melakukan exploit, dilakukannya brute force pada layanan yang terbuka pada target. Brute force attack adalah sebuah metode untuk menebak suatu password dari sebuah enkripsi atau sebuah otentikasi dengan cara mencobanya berkali-kali

dengan berbagai macam kombinasi huruf, angka dan simbol. Tool yang digunakan yaitu Hydra. *Command* yang digunakan adalah :

#Hydra -l username –P password.list [ip_target] [nama_service]

- **-l** digunakan jika kita mempunyai list user yg kemungkinan dipake (ga pake -L jg bisa kok tapi harus di sebut usernya, tapi harus huruf L kecil, contoh –> -l admin).
- **-P** adalah wordlist yang kita punya.
- **Ip_target** diisi target kita, bisa webserver, router, switch, apapun itu asal ada ip dan service yg berjalan (dan termasuk dalam service yg ada di hydra tentunya)
- **service** adalah sesuai yang ada di help hydra.

SIMULASI EKSPLORASI WEB MENGGUNAKAN WEB GOAT

OWASP (*The Open Web Application Security Project*) @mempunyai visi ingin menginformasikan bahwa web application pada dasarnya tidak aman. OWASP *Vulnerable Web Applications Directory* (VWAD) adalah *registry* komprehensif dan semua aplikasi web yang dikenal rentan. Aplikasi web yang rentan tersbut dapat digunakan oleh pengembang web, auditor keamanan dan penguji penetrasi untuk dimasukkan ke dalam praktek pengetahuan dan keterampilan mereka selama sesi pelatihan (dan terutama setelah itu), serta untuk menguji setiap saat hacking tools berganda dan teknik ofensif tersedia, dalam persiapan untuk selanjutnya keterlibatan mereka di dunia nyata. Tujuan dibuatnya aplikasi tersebut adalah sebagai *framework* untuk menemukan kelemahan aplikasi web.

Resiko-resiko keamanan aplikasi OWASP, yaitu:

A1 – Injeksi	Kelemahan injeksi, seperti injeksi SQL, OS, dan LDAP, terjadi ketika data yang tidak dapat dipercaya dikirim ke suatu interpreter sebagai bagian dari suatu perintah atau query. Data berbahaya dari penyerang tersebut dapat mengelabui interpreter untuk mengeksekusi perintah yang tidak direncanakan, atau untuk mengakses data yang tidak terotorisasi.

A2 – Cross-Site Scripting (XSS)	Kelemahan XSS terjadi ketika aplikasi mengambil data yang tidak dapat dipercaya dan mengirimnya ke suatu web browser tanpa validasi yang memadai. XSS memungkinkan penyerang mengeksekusi script-script di dalam browser korban, yang dapat membajak sesi pengguna, mengubah tampilan website, atau mengarahkan pengguna ke situs-situs jahat.
A3 – Otentikasi dan Pengelolaan Sesi yang Buruk	Fungsi-fungsi aplikasi yang berhubungan dengan otentikasi dan pengelolaan sesi seringkali tidak dimplementasikan dengan benar. Hal ini memungkinkan penyerang mendapatkan password, key, dan token-token sesi, atau mengeksploitasi cacat implementasi lainnya untuk memperoleh identitas pengguna yang lain.
A4 –Referensi Obyek Langsung yang Tidak Aman	*Direct object reference* terjadi ketika pengembang mengekspos referensi ke suatu objek implementasi internal, seperti file, direktori, atau kunci database. Tanpa adanya suatu pemeriksaan kendali akses atau perlindungan lainnya, penyerang dapat memanipulasi referensi-referensi ini untuk mengakses data yang tidak terotorisasi.

A5 – Cross-Site Request Forgery (CSRF)	Suatu serangan CSRF memaksa browser korban yang sudah log-on untuk mengirim HTTP request yang dipalsukan, termasuk di dalamnya session cookie korban dan informasi otentikasi lain yang otomatis disertakan, ke suatu aplikasi web yang rentan. Hal ini memungkinkan penyerang untuk memaksa browser korban menghasilkan request yang dianggap sah oleh aplikasi rentan tadi.
A6 – Kesalahan Konfigurasi Keamanan	Keamanan yang baik mensyaratkan dimilikinya suatu konfigurasi keamanan (yang terdefinisi dan diterapkan) untuk aplikasi, framework, server aplikasi, web server, server database, dan platform. Semua pengaturan ini harus didefinisikan, diimplementasikan,dan dipelihara, karena terdapat banyak aplikasi yang dirilis tanpa konfigurasi default yang aman. Hal ini juga mencakup menjaga semua software up-to-date, termasuk semua pustaka kode yang digunakan aplikasi tersebut.
A7 – Penyimpanan Kriptografi yang Tidak Aman	Banyak aplikasi web yang tidak melindungi data sensitif (seperti data kartu kredit, SSN, kredensial otentikasi) dengan enkripsi atau hashing yang memadai. Penyerang dapat mencuri atau memodifikasi data dengan perlindungan lemah semacam itu untuk melakukan pencurian identitas, kejahatan kartu kredit, atau kriminalitas lain.

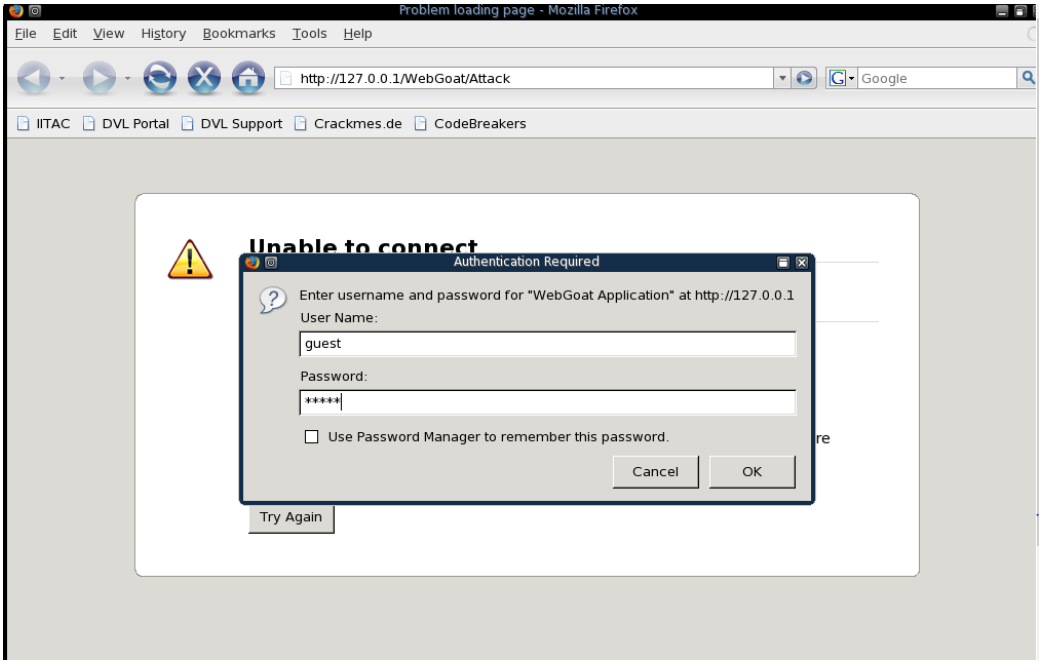
A8 – Kegagalan Membatasi Akses URL	Banyak aplikasi web memeriksa hak akses URL sebelum memberikan link dan tombol-tombol yang diproteksi. Bagaimanapun juga, aplikasi perlu melakukan pemeriksaan kendali akses yang serupa setiap kali halaman-halaman ini diakses, atau penyerang akan dapat memalsukan URL untuk mengakses halaman-halaman yang tersembunyi ini,
A9 – Perlindungan yang Tidak Cukup pada Layer Transport	Aplikasi seringkali gagal untuk mengotentikasi, mengenkripsi, dan melindungi kerahasiaan serta integritas lalu-lintas jaringan yang sensitif. Ketika aplikasi gagal melakukan hal-hal tersebut, adalah dikarenakan ia mendukung algoritma yang lemah, menggunakan sertifikat yang tidak valid atau sudah kadaluarsa, atau karena tidak menggunakannya dengan benar
A10 – Redirect dan Forward yang Tidak DIvalidasi	Aplikasi web seringkali mengarahkan (redirect) dan meneruskan (forward) pengguna ke halaman dan website lain, dan mengunakan data yang tidak dapat dipercaya untuk menentukan halaman tujuan. Tanpa validasi yang tepat, penyerang dapat mengarahkan korban ke situs phishing atau malware, atau menggunakan forward untuk mengakses halaman yang tidak terotorisasi.

Pengujian target eksploitasi yang dilakukan yaitu dalam linkungan *local network.* Aplikasi yang digunakan merupakan WebGoat yang terdapat pada DVL, url: 127.0.0.1/WebGoat/attack.

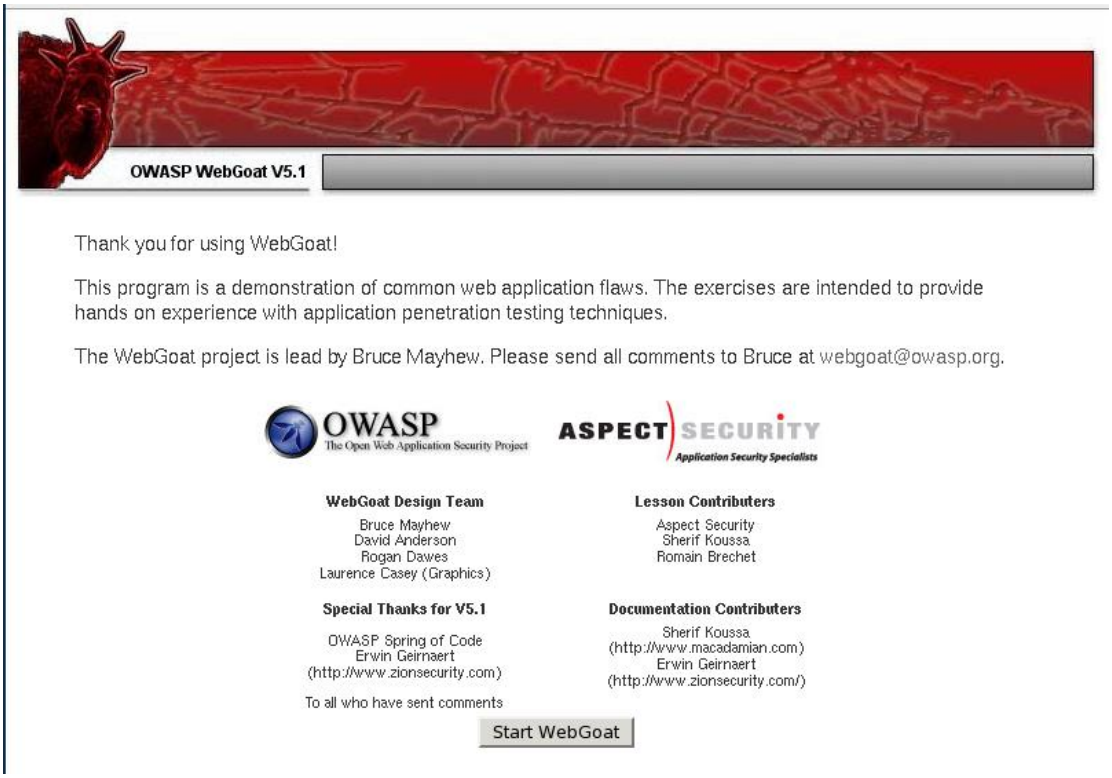
Berikut adalah langkah-langkahnya :

- Startx di DVL untuk masuk ketampilan GUI.
- Jalankan WebGoat (lihat gambar 7).

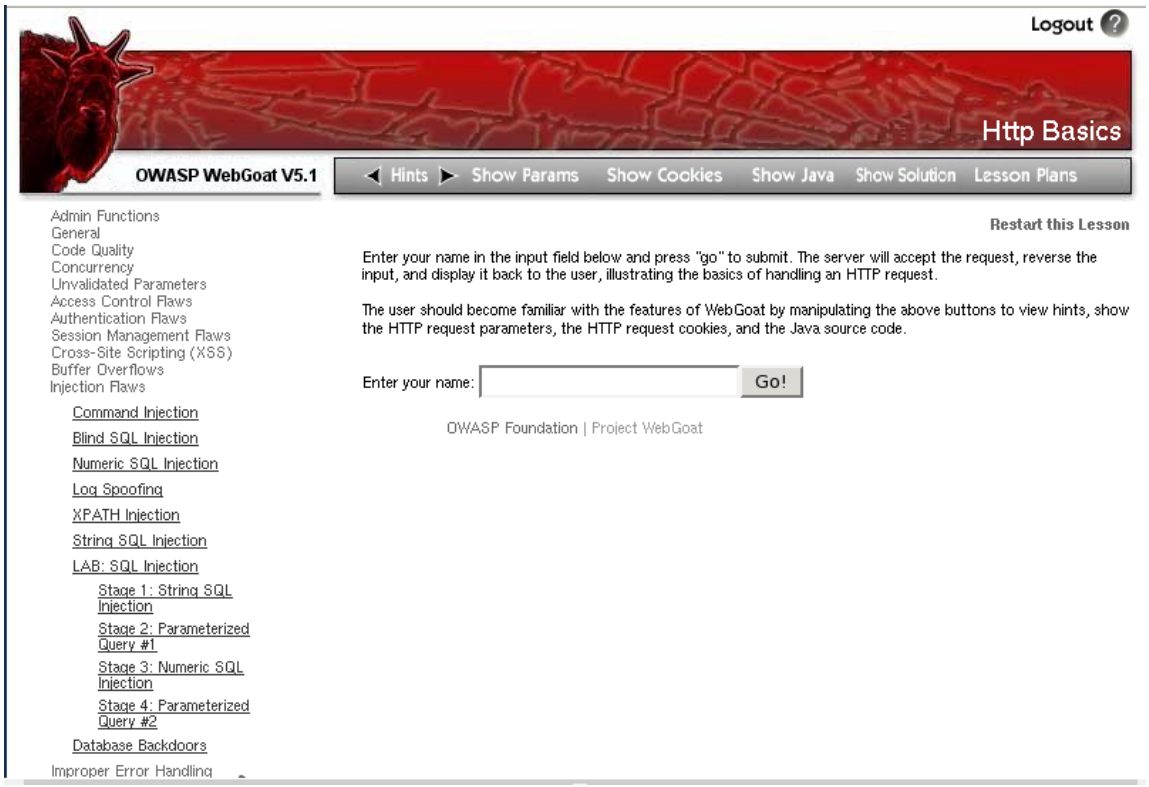
Gambar 7: proses menjalankan WebGoat



Gambar 8: masukan *username* dan *password* untuk sign in WebGoat

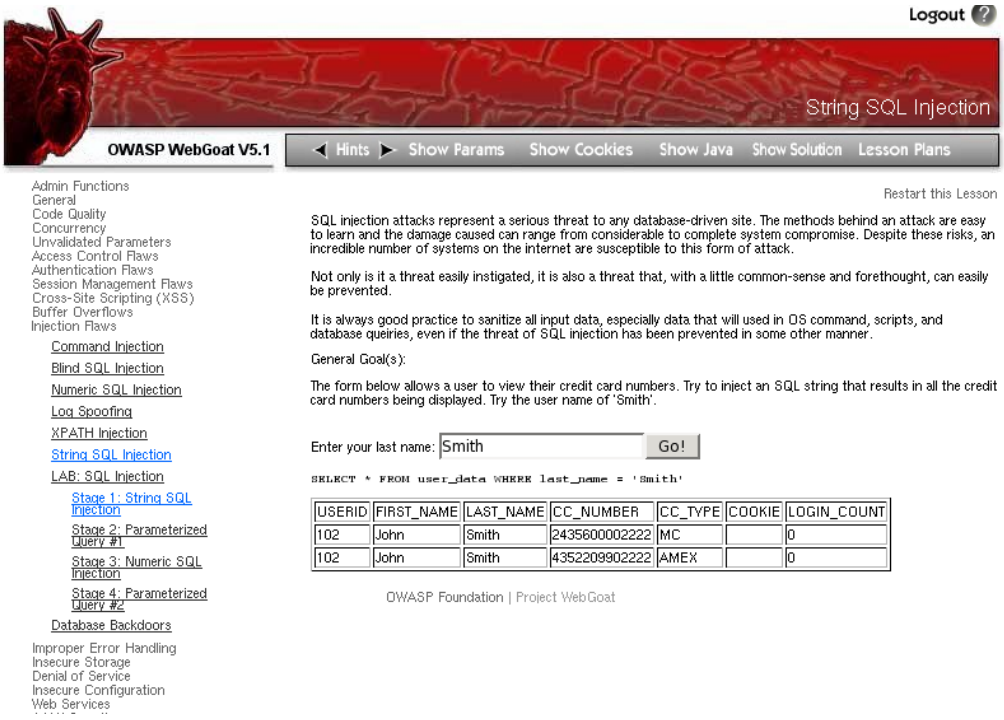


Gambar 8: *interface* WebGoat (langsung saja klik Start WebGoat)



Gambar 9: interface WebGoat lanjutan

Pada aplikasi WebGoat akan dilaksanakan pengujian dengan simulasi blind SQL injection. Pada percobaan tersebut dilakukannya pencarian semua nama dengan last name “Smith” atau user name Smith pada tabel “user_data” yang terdapat dalam WebGoat.



Gambar 10: hasil blind SQL injection, “Smith”

Didapatkannya tabel “user_data” tersebut karena string SQL Injection, tidak melakukan filter input yang masuk.

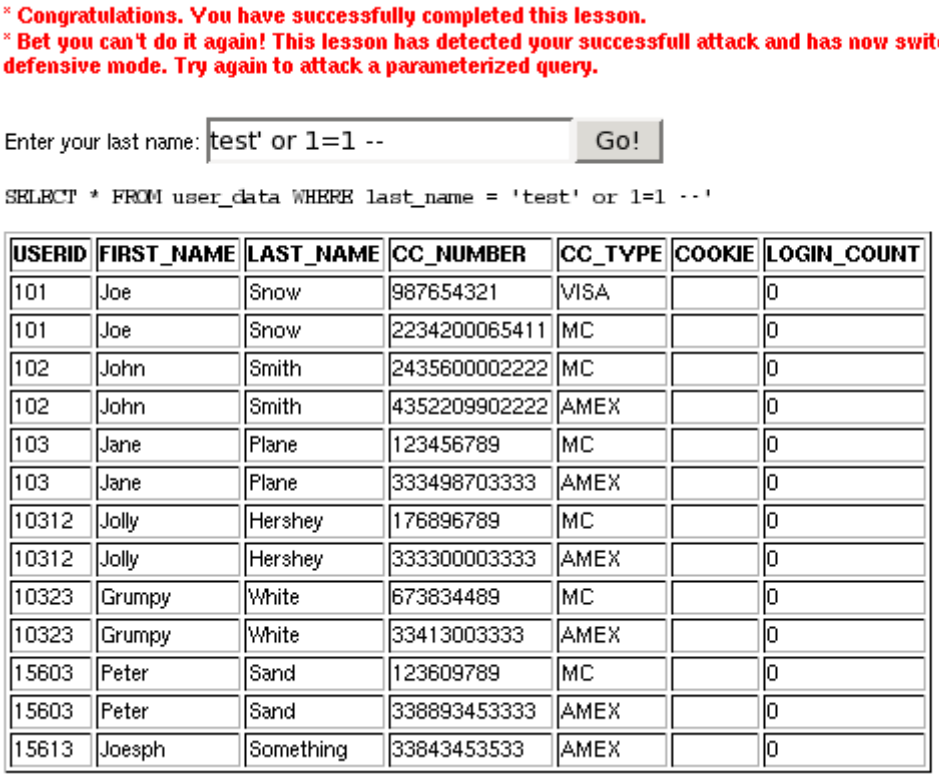
Untuk melakukan skenario serangan, aplikasi menggunakan data yang tidak dapat dipercaya dalam kontuksi *SQL call* yang rentan adalah sebagai berikut:

String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") +"'";

atau dengan,

‘1’=’1

Pada kasus diatas enyerang memodifikasi parameter 'id' dalam browser mereka untuk mengirim:‘ or '1'='1. Hal tersebut mengubah arti *query* untuk mengembalikan semua *record* database akun.



* **Congratulations. You have successfully completed this lesson.**
* **Bet you can't do it again! This lesson has detected your successfull attack and has now switched to a defensive mode. Try again to attack a parameterized query.**

Enter your last name: test' or 1=1 -- Go!

```
SELECT * FROM user_data WHERE last_name = 'test' or 1=1 --'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		0
10312	Jolly	Hershey	333300003333	AMEX		0
10323	Grumpy	White	673834489	MC		0
10323	Grumpy	White	33413003333	AMEX		0
15603	Peter	Sand	123609789	MC		0
15603	Peter	Sand	338893453333	AMEX		0
15613	Joesph	Something	33843453533	AMEX		0

Gambar 11: simulasi melakukan penyerangan

Pada gambar 11 tersebut, pertama-tama kita men*select all* dari table “user_data”. Awalnya akan ditambahkan tanda petik dan akan membaca last name yg kita masukkan, maksud 1=1 adalah prinsip aljabar *boolean true*, dimana walaupun kita salah masih akan bernilai *true*. Itulah kesalahan dari program karena tidak memfilter terlebih dahulu.

DAFTAR PUSTAKA

[1] Owasp, "OWASP Top 10 - 2010 Versi Indonesia," *OWASP Top 10 Versi Indones.*, 2010.

[2] A. Sanmorino, "SIMULASI EKSPLORASI WEB MENGGUNAKA NW3AF DAN WEB GOAT," vol. 7, no. 1, 2016.