ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

SIMULASI EKSPLORASI WEB MENGGUNAKA NW3AF DAN WEB GOAT SERTA ALTERNATIF PENCEGAHANNYA

Ahmad Sanmorino[1)]

[1)] *Program Studi Sistem Informasi Universitas Indo Global Mandiri*
Jl. Jend. Sudirman No. 629 KM.4 Palembang Kode Pos 30129
Email : sanmorino@uigm.ac.id[1)]

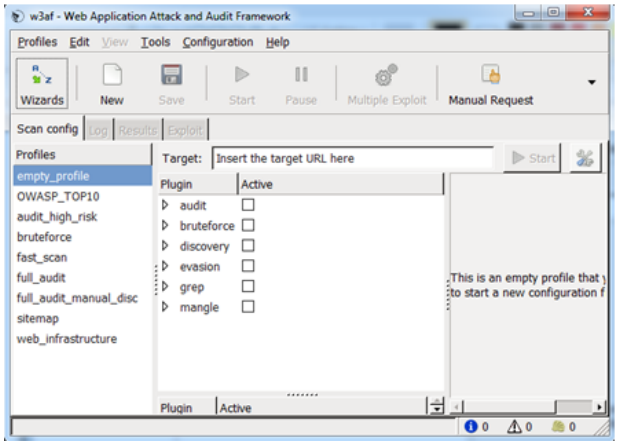
ABSTRACT

Exploration web is the means used to obtain information on a web page specifically related to security issues. In this research, exploratory simulation tool W3AF danWebGoat.W3AF web use is short for Web Application Attack and Audit Framework. The objective of this application is as a framework to find the weakness of web applications. While Web Goat is a web page that is targeted for exploration on the environment localhost. Simulation gives positive results of various web pages of information targeted exploration. Among the security loopholes that can be used as roads by the cracker to commit an illegal act. Alternative solutions provided in order to close the security gap, so as to minimize the risk r a crime that may occur.

Key words *: web exploration, W3AF, webgoat*

1. Pendahuluan

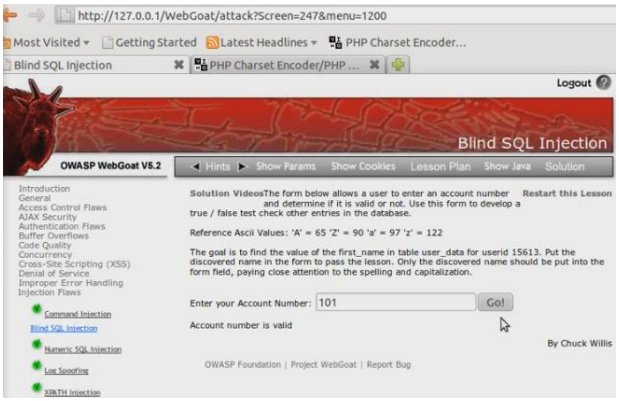
W3AF adalah singkatan dari Web Application Attack and Audit Framework [1]. Tujuan dibuatnya aplikasi ini adalah sebagai framework untuk menemukan kelemahan aplikasi web. Adapun antarmuka aplikasi w3af dapat dilihat pada gambar 1. Dengan menggunakan W3af, user diberikan banyak pilihan untuk mengekplorasi suatu halaman web[2]. User dapat memilih profile sesuai dengan keinginannya, mengeksplorasi kelemahan suatu website hanya dengan mencentang check box yang disediakan. Tentu saja semakin banyak jenis kelemahan yang ingin diketahui, semakin lama waktu yang dibutuhkan untuk melakukan eksplorasi.



Gambar 1. *Antarmuka W3AF*

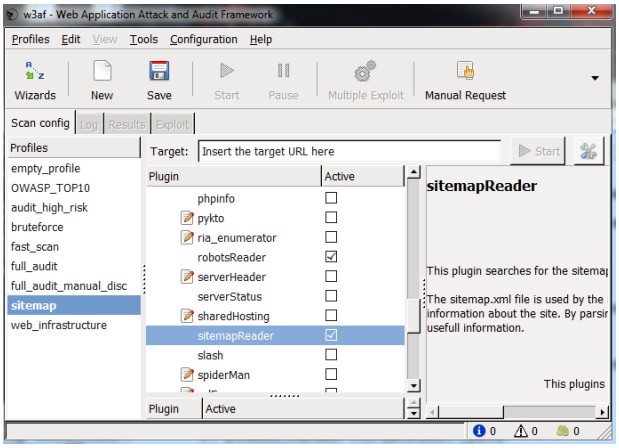
A. Pengujian

Pada pengujian pertamaini yang menjadi target eksploitasiadalahaplikasi web dalamlingkunganlocal network. Aplikasi yang digunakan adalah webgoat dengan alamat url : 127.0.0.1/WebGoat/attack [3].



Gambar 2. *Antarmuka Aplikasi Webgoat*

Adapun profile w3af yang penulis pilih adalah sitemap dengan plugin Robots Reader dan Sitemap Reader.

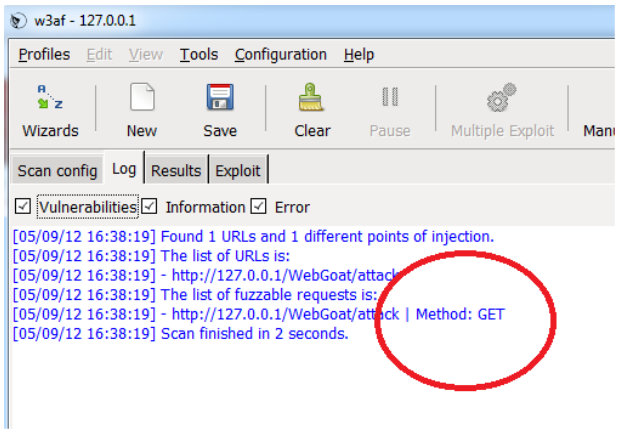


Gambar 3. *Profile w3af*

Pada gambar 4 diperlihatkan hasil scan aplikasi webgoat menggunakan w3af. Sama seperti aplikasi web pada

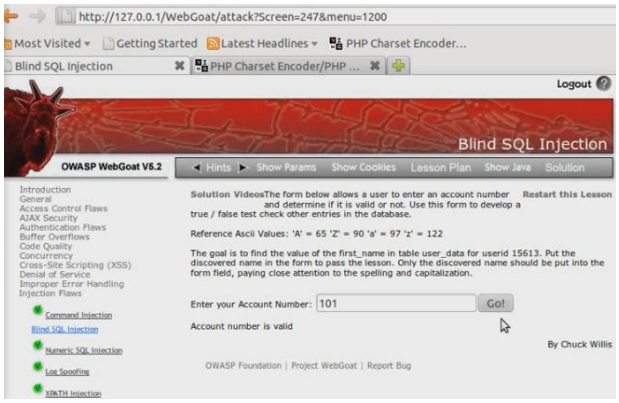
ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

umumnya, kesalahan yang paling sering ditemukan adalah penggunaan method GET. Mengapa penggunaan method GET ini dikatakan tidak tepat, Jawaban dari pertanyaan ini akan penulis bahas pada bagian pembahasan.



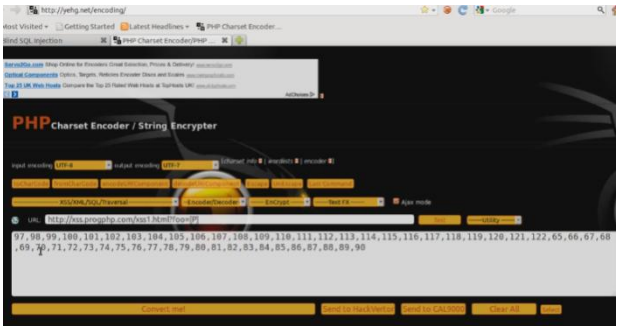
Gambar 4. *Hasil Scan Webgoat Menggunakan W3AF*

Selanjutnya untuk pengujian kedua penulis akan melakukan simulasi blind SQL injection. Penulis masih menggunakan tool yang sama yaitu webgoat. Gambar 5 memperlihatkan halaman lesson blind SQL injection yang disediakan webgoat [3].



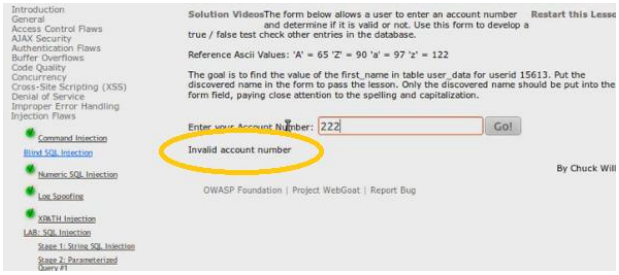
Gambar 5. *Blind SQL Injection* pada Webgoat

Tujuan blind SQLi yang dilakukan adalah untuk mendapatkan nilai dari atribut "first_name" dalam tabel "user_data" dan untuk userid = 15613. Sebagai referensi diketahui nilai ASCII 'A' = 65; 'Z' = 90; 'a' = 97; 'z' = 122. Untuk memudahkan dalam mengubah karakter menjadi nilai ASCII dapat digunakan PHP Charset Encoder tool yang terdapat pada website pengkodean yehg [4]. Misalnya jika dimasukkan baris karakter: "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ" Maka akan ditampilkan hasil seperti pada gambar 2 dibawah ini:



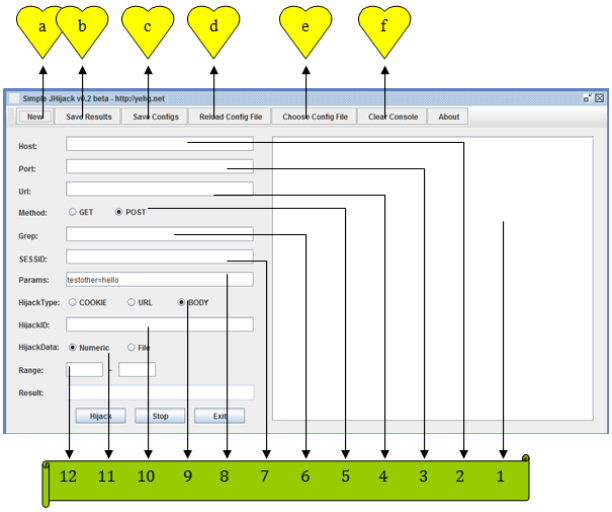
Gambar 6. *PHP Charset Encoder*

Selanjutnya yang akan dilakukan adalah usaha untuk mendapatkan nilai ASCII setiap karakter first_name lalu mengubahnya menjadi karakter. Kemudian memasukkannya kedalam form yang telah disediakan. Kita bisa mencoba memasukkan sembarang karakter (berupa huruf atau angka) kedalam form yang disediakan, jika nilai first_name yang dimasukkan tidak sesuai dengan nilai yang ada dalam database akan ditampilkan pesan "invalid account numb" (gambar 7).



Gambar 7. *Pesan yang Ditampilkan*

Salah satu cara yang dapat digunakan untuk mendapatkan nilai ASCII setiap karakter first_name dalam tabel user_data adalah dengan menggunakan tool Jhijack (http://yehg.net).



Gambar 8. *Tampilan Aplikasi Jhijack*

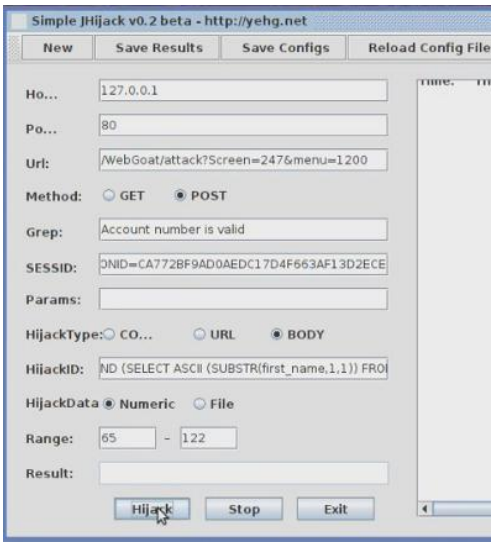
ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

Informasi pada gambar 4 adalah sebagai berikut:

a. Tab new digunakan untuk mengosongkan form atau dengan kata lain membuat konfigurasi yang baru.
b. Tab save results digunakan untuk menyimpan hasil hijack yang berhasil dilakukan.
c. Tab save configs digunakan untuk menyimpan konfigurasi yang sedang berjalan.
d. Tab reload config file digunakan untuk memakai kembali konfigurasi yang telah dipilih sebelumnya.
e. Tab choose config file digunakan untuk memilih konfigurasi yang pernah disimpan sebelumnya.
f. Tab clear console digunakan untuk membersihkan console.

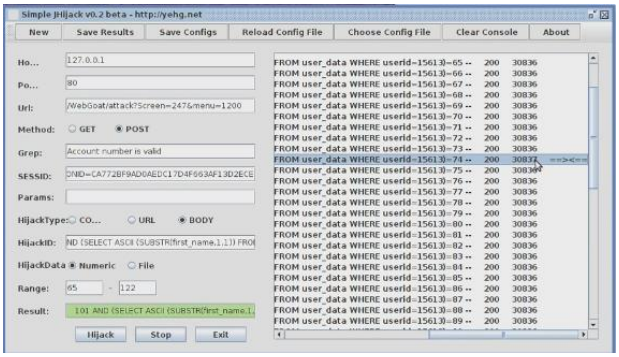
1. Console tempat menampilkan hasil hijack
2. Host yang digunakan, bisa berupa alamat lokal/localhost (127.0.0.1).
3. Port yang digunakan, default-nya port 80
4. URL, path setelah host pada alamat website, bisa berupa page, parameter atau value. Pada pengujian ini digunakan
5. URL:/WebGoat/attack?Screen=247&menu=1200
6. Method yang digunakan untuk mendapatkan nilai berdasarkan query yang dimasukkan. Terdapat dua pilihan method yaitu GET dan POST.
7. Nilai yang dikembalikan ketika query berhasil dilakukan. Karena kita mencari nilai ASCII yang benar (valid) maka nilai grep digunakan "Account number is valid".
8. 7. SESSIONID diisi dengan nilai session berdasarkan nilai yang dikembalikan oleh cookies yang digunakan oleh webgoat. Nilai session_id berupa nilai yang unik, misalnya JSESSIONID= 2F20EE6A5DBE999D35F657160A10D9
9. Nilai parameter hanya berupa opsional, karena tidak mempengaruhi hasil query yang digunakan.
10. HijackType, tipe hijack yang digunakan. Terdapat 3 pilihan tipe yaitu COOKIE, URL dan BODY. Karena vurnerabilitas yang akan dieksplorasi terdapat pada body script maka digunakan tipe BODY.
10. HijackID diisi dengan perintah query (SQL) untuk mendapatkan nilai ASCII karakter atribut first_name pada tabel user_data.
11. HijackData, pilihan data yang dihasilkan. Terdapat dua pilihan yaitu numeric dan file. Karena hasil dari hijack yang dilakukan berupa angka maka dipilih numeric.
12. Untuk nilai range, sesuai dengan referensi nilai ASCII yang diberikan webgoat yaitu 65 untuk 'A' hingga 122 untuk 'z'.

Sehingga setelah seluruh nilai dimasukkan, tampilan Jhijack menjadi seperti pada gambar 8. Untuk mulai melakukan hijack tekan button Hijack:



Gambar 9. *Jhijack dengan nilai-nilai yang telah dimasukkan*

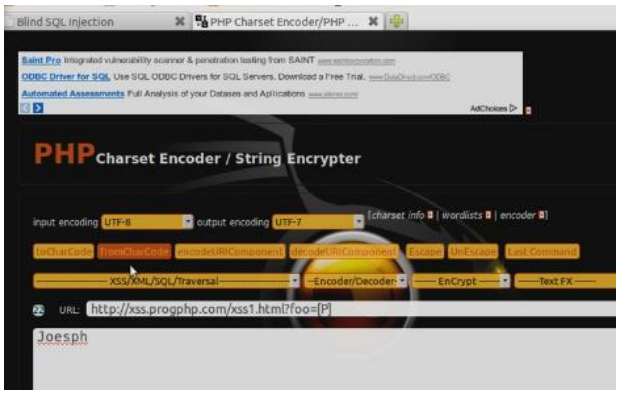
Apabila tidak terdapat kesalahan, Jhijack akan menampilkan hasil query seperti pada gambar 9. Dapat dilihat karakter pertama dari first_name adalah nilai ASCII 74.



Gambar 10. *Hasil Hijack yang telah dilakukan*

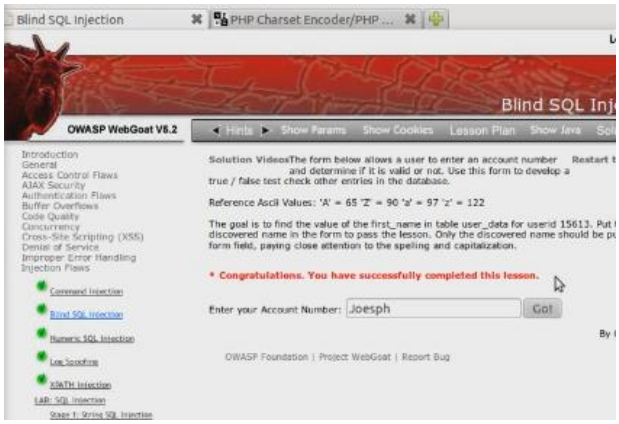
Penulis melakukan hal yang sama untuk mendapatkan nilai ASCII karakter kedua, ketiga dan seterusnya hingga diperoleh karakter first_name secara keseluruhan. Berdasarkan hijack yang telah dilakukan, diperoleh hasil akhir nilai ASCII atribut first_name adalah 74, 111, 101,115, 112, 104. Untuk mendapatkan karakter atribut first_name kembali kita menggunakan PHP charset encoder (Gambar 10).

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786



Gambar 11. *Penggunaan PHP Charset Encoder*

Ternyata setelah diubah menjadi karakter, diperoleh nilai untuk first_name = "Joesph". Sama seperti sebelumnya, kita masukkan kata "Joesph" ini kedalam form yang terdapat pada halaman blind SQL injection webgoat. Jika nilai *first_name* yang dimasukkan benar, akan ditampilkan pesan "congratulations" seperti yang diperlihatkan gambar 11.



Gambar 12. *Pesan Congratulations*

2. Pembahasan

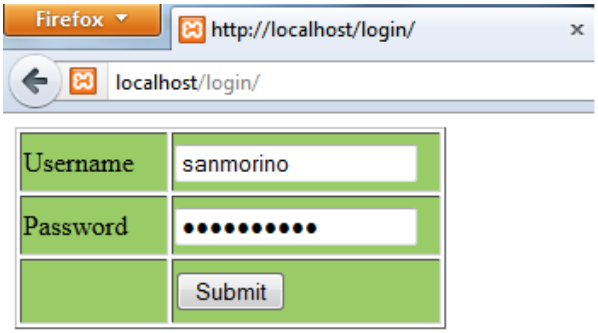
Pada bagian ini peneliti akan membahas pengujian yang telah dilakukan. Untuk pengujian pertama peneliti akan memberikan contoh sederhana penggunaan method GET. Langkah-langkahnya adalah sebagai berikut :

1. Buat aplikasi login yang dapat menerima masukkan berupa username dan password dari user, bahasa yang digunakan adalah PHP.
2. Source dari aplikasi login adalah seperti terlihat pada gambar 5. Pada gambar terlihat digunakan method GET.

```
<form name="login" method="get" action="tampil.php">
  <table width="253" height="118" border="1">
    <tr bgcolor="#99CC66">
      <td width="83">Username</td>
      <td width="154"><input name="username" type="text" id="username">
    </tr>
    <tr bgcolor="#99CC66">
      <td>Password</td>
      <td><input name="password" type="text" id="password"></td>
    </tr>
    <tr bgcolor="#99CC66">
      <td> </td>
      <td><input type="submit" name="Submit" value="Submit"></td>
    </tr>
  </table>
</form>
<?php
echo $_GET['username'];
echo $_GET['password'];
?>
```

Gambar 13. *Source Code Aplikasi Login Menggunakan Method GET*

3. Selanjutnya jalankan melalui browser (lingkungan localhost), masukkan username dan password.



Gambar 14. *Antarmuka Aplikasi login*

4. Selanjutnya lihat hasil (output) yang ditampilkan ketika menggunakan method GET. Apa yang terjadi?

Gambar 15. *Output Aplikasi Login*

5. Aplikasi menampilkan nilai Username dan Password yang dimasukkan user pada bagian URL. Hal ini akan sangat beresiko ketika aplikasi web tersebut menyangkut data-data penting, seperti internet banking, inteligen, dan lain-lain. Oleh karena itu, perlu ada solusi yang dapat digunakan untuk menggantikan tugas method GET.

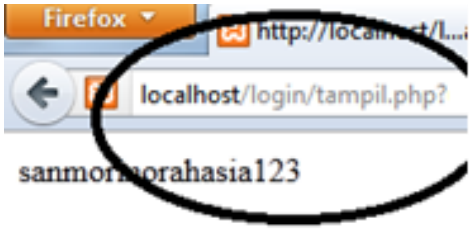
Alternatif solusi yang dapat diberikan yaitu **langkah pertama** adalah penggunaan method POST. Sehingga *source code* aplikasi login menjadi :

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

```
<form name="login" method="post" action="tampil.php">
  <table width="253" height="118" border="1">
    <tr bgcolor="#99CC66">
      <td width="83">Username</td>
      <td width="154"><input name="username" type="text" id="username"><
    </tr>
    <tr bgcolor="#99CC66">
      <td>Password</td>
      <td><input name="password" type="password" id="password"></td>
    </tr>
    <tr bgcolor="#99CC66">
      <td> </td>
      <td><input type="submit" name="Submit" value="Submit"></td>
    </tr>
  </table>
</form>
<?php
echo $_POST['username'];
echo $_POST['password'];
?>
```

Gambar 16. *Source Code Aplikasi Login Menggunakan Method POST*

Ketika dijalankan pada browser (lingkungan localhost), output yang ditampilkan seperti pada gambar 9. Nilai username dan password tidak lagi di tampilkan dibagian URL.



Gambar 17. *Output Aplikasi Login*

Solusi pencegahan adanya celah keamanan yang telah dibahas merupakan salah satu alternatif yang sederhana namun dinilai cukup mumpuni sebagai benteng terluar untuk meminimalisir berbagai tindakan ilegal.

Selanjutnya untuk pengujian keduaya itu serangan blind SQL Injection. Serangan blind SQL Injection terjadi diakibatkan oleh tanda petiksatu yang diloloskan kedalam query *database* sehingga mengakibatkan database akan memproses tanda petik satu sebagai bagian dari query yang nantinya akan menghasilkan error. Untuk mengatasinya, pada konfigurasi php.ini pastikan tag

m_quote_gpc = on

Biasanya, instalasi php standar di beberapa system operasilinux, tag

m_quote_gpc = off

adalah sebagai *default*. Untuk itu harus dirubah menjadi on. Dengan demikian, tanda petikakan diberikan *escape character* di depannya.

Langkah kedua adalah menyembunyikan pesan error. Pesan error diibaratkan peta menuju tempat harta karun bagi penyerang. Pesan error ditampilkanhanya pada saat development saja. Sedangkan ketika akan di *publish*, Pesan error sama sekali tidak boleh ditampilkan, yaitu dengan menambahkan fungsi

error_reporting(0);

Langkah ketiga adalah menggunakan framework untuk standarisasi kode. Dengan menggunakan framework kode/script akan lebih rapi dan keamanannya juga sudah standar. Penggunaan framework juga akan diikuti dengan keharusan mengaktifkan *module* mod_rewrite pada apache webserver danjugamengaktifkan*module* curl pada php yang akan membuat serangan SQL Injection menjadi tidak dapat dilakukan.

Langkah terakhir adalah membuat password admin yang kompleks. Serangan blind SQL Injection intinya adalah berusaha melihat isi table dimana informasi *username* dan *password* admin disimpan. Dengan demikian, penyerangakan login sebagai admin kehalaman administrasi. Katakanlah cara-cara diatas berhasil ditembus, admin masih mempunyai pertahanan terakhir yaitu pada level password. Password haruslah disimpan dalam bentuks udah terenkripsi. Sehingga seandainya penyerang bias melihat isi username dan password, mereka tetap tidak bisa login, karena passwordnya terenkripsi. Kita bias menggunakan algoritma MD5 yang merupakan metode enkripsi satuarah [5]. Dimana bisa di *encrypt* tapi tidak bisa di *decrypt*. Meskipun demikian, password yang kita gunakan juga haruslah tidak mudah ditebak karena di internet juga ada banyak web yang menyediakan layanan decrypt MD5 yang sebenarnya cara kerjanya tidaklah mendecrypt tetapi mencari kecocokan atau kesamaan dengan yang ada di database yang mereka miliki.

3. Kesimpulan

Dengan meminimalisir celah keamanan, yanga da di halaman web, dapat mengurangi resiko terjadinya tindakan ilegal yang dilakukan *attacker*. Simulasi yang telah dilakukan telah berhasil membuk tikan tingginya resiko tindakan ilegal yang dapat terjadi pada halaman web. Hanya dengan menggunakan tool sederhana, seseorang yang tidak memiliki keahlian dalam programming pun dapat dengan mudah memanfaatkan celah keamanan hasil eksplorasi pada halaman web. Solusi pencegahan yang diberikan dapat menjadi alternative sebagai benteng terluar dalamme lindungi halaman web dariberbagai serangan yang dilakukan *attacker*.

Daftar Pustaka

[1] http://w3af.org/
[2] Vibhandik, R., "Vulnerability assessment of web applications – a testing approach," *in Proc. IEEE 2015 Forth International Conference on e-Technologies and Networks for Development (ICeND)*, pp. 1 – 6, Sept. 21-23, 2015.
[3] https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

ISSN PRINT : 2302-500X
ISSN ONLINE : 2477-3786

[4] http://yehg.net/encoding/
[5] Z. Yong-Xia and Z. Ge, "MD5 Research," in Proc. IEEE 2010 Second International Conference on Multimedia and Information Technology (MMIT), pp. 271-273, April 24-25, 2010.