ICSIIT 2012-1

by Agustinus Noertjahyana

Website Application Security Scanner Using Local File Inclusion and Remote File Inclusion

Agustinus Noertjahyana
Petra Christian University
Siwalankerto 121-131
Surabaya
+6231-2983456
agust@petra.ac.id

Ibnu Gunawan
Petra Christian University
Siwalankerto 121-131
Surabaya
+6231-2983456
ibnu@petra.ac.id

Deddie Tjahjono
Petra Christian University
Siwalankerto 121-131
Surabaya
+6231-2983456

ABSTRACT

Today many web-based applications developed to be accessible via the internet. The problem that often occurs is commonly found on web application vulnerabilities. Many application developers often ignore security issues when developing applications that can cause substantial losses if a hacker manages to gain access to the system. A hacker can replace web pages, obtain sensitive information, or even take over control of the website.

For that reason there is a need for applications that can help developers to overcome these problems. This application is expected to detect the vulnerabilities that exist on a website.

Existing processes include: The process of crawling to get the whole link from target websites, attacking process is useful for testing the attacks, and the last is the process of displaying a report about the security hole on the website. This application is developed using Microsoft Visual C # 2010.

Based on the results of tests made on this application, it can be concluded that the application can detect vulnerabilities in the website and report any form of link that has a security hole on the website.

Keywords

Website Application, Security Scanner, Vulnerability scanner.

1. INTRODUCTION

System security is a priority for a web administrator and web developer. But in fact web developers still prefer the design and interesting topics to increase the traffic on the website. Safety on the website is often overlooked. Though the website application security is the most important because of the presence of vulnerabilities on the website then the website will be attacked by hackers [3].

The purpose of such attacks include: extract data from websites such important credit card information / customer data, defacing the website, or even a hacker to take control of the websites server.

Based on the 2009 report from WASC (Web Application Security Consortium), an organization that conducts research in the field of web application security, attacks on the application site is increasing every year. In the report mentioned that until the end of 2009, 87% of the total existing websites still have vulnerability that can be fatal to the application site [6]. The impact of the vulnerability is not only detrimental to the developer but also detrimental to the user. Application on the website easily attacked because applications usually have different flaws and easily exploited. To help web developers in addressing these issues, it would require an application Web Application Security Scanner to detect security holes in the website and produce a report that provides a security hole on the website automatically [5].

2. WEBSITE APPLICATION

2.1 Website Application Layer

Application layer is the layer that connects a website to its users around the world [1]. At this layer there is a database containing sensitive information like credit card number, name, address, birthdate, financial records, trade secrets, medical data, and more.

At this layer is usually the main attack for hackers who have a variety of purposes, such as information theft, damage the website, or even take over control of the website. This layer was attacked because it is the weakest layer than other layers. The rapid development of technology can make more complex the application layer. There will be lots of bugs and the possibility of security holes that can be exploited by hackers to get into the system.

From the report WASC (Web Application Security Consortium), said that currently 75% of cyber attacks in the world begins at the application layer and can adversely impact on the website [6].

2.1.1 Website Application Security

Web application security is a way to protect the site from the application layer attacks that can cause losses to a variety of individual and corporate [7].

2.1.2 Website Application Security Scanner

Web Application Security Scanner is the software that automatically searches for vulnerabilities that exist on the website. This software does not access the source code in carrying out its action, but using the Black Box [5].

Black Box is a step in which the application will perform experiments to attack and then capture the response of the attempted attacks have been carried out, after which the application will process and evaluate the response to a report in the form of an overview of security loopholes that exist on the website.

This method is commonly known as Penetration Testing.

These applications are generally divided into three stages in carrying out its action, namely Crawling Component, Component Attack, and Analysis Modules [5].

- Crawling Component, which is popular with the term Web Crawler, the website will index the links on the website. This step can be performed by various methods, depending on the needs of application users.
- Attack Component, in which the application will start automatic attempted attacks on a link that has been indexed.
- Analysis Modules in which the application will evaluate the responses provided by the website and create reports of website security picture.

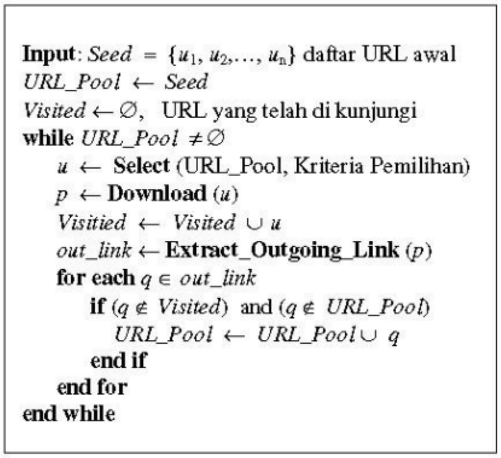
2.2 Web Crawler

Web crawlers are also commonly known as the Spider Web. This method collects all the information on web pages. Web Crawler log every link on the website pages visited, and then browse through these links one by one. In its implementation, there are various methods of web crawlers that are used as required [7].

The methods commonly used on the Web Crawler are:

- BFS (Breadth First Search)
  Search based on the breadth of website information, on the implementation uses queue as the storage of the URL
- DFS (Depth First Search)
  Search depth information on the website, on the implementation uses stack as the storage of the URL

A Web crawler process generally begins with providing the initial set of URLs as the beginning of a search into a queue. Priority of the criteria can be applied to reorder the list of URLs in the queue. Furthermore Crawler downloads pages by URL from the queue. Once deposited into the collection, obtained



```
Input: $Seed = \{u_1, u_2, \ldots, u_n\}$ daftar URL awal
$URL\_Pool \leftarrow Seed$
$Visited \leftarrow \emptyset$,  URL yang telah di kunjungi
while $URL\_Pool \neq \emptyset$
    $u \leftarrow$ Select (URL_Pool, Kriteria Pemilihan)
    $p \leftarrow$ Download ($u$)
    $Visitied \leftarrow Visited \cup u$
    $out\_link \leftarrow$ Extract_Outgoing_Link ($p$)
    for each $q \in out\_link$
        if ($q \notin Visited$) and ($q \notin URL\_Pool$)
            $URL\_Pool \leftarrow URL\_Pool \cup q$
        end if
    end for
end while
```

Figure 1 Generic Web Crawler Algorithm

2.3 File Inclusion

File Inclusion is a method for attacker to insert malicious code into a site that has security holes, where penetration is done by two ways:

- Local File Inclusion (LFI)
- Remote File Inclusion (RFI)

2.3.1 LFI (Local File Inclusion)

Local File Inclusion is a web application flaws that led to an attacker could exploit the Web Server by accessing the directory and execute the command in web server other than root. [2].

Local File Inclusion closely related to the Directory Traversal. LFI performed locally from the web server. LFI normally occurs due to the variables during of website creation is poorly controlled thus allowing an attacker to access the folder directly.

Functions which could lead to Local File Inclusion (LFI) on the website are:

- include ();
- include_once ();
- require ();
- require_once ();

With the condition in the php configuration on the server:

- allow_url_include = on
- allow_url_fopen = on
- magic_quotes_gpc = off

Example code:

```
<? php
$ page = $ _GET ['page'];
include ($ page);
?>
```

If the function is accessed by the LFI methods of:

- http://victim.com/index.php?page=../../../../../../../etc/passwd

It will display the contents of / etc / passwd, where '.. /' is a method to move one folder up from the current folder. Development of the LFI method is Remote Connect-Back Shell, where an attacker gains a shell of the target server and the attacker could perform a variety of linux commands on the existing system.

2.3.2 RFI (Remote File Inclusion)

Remote File Inclusion is the type of attack that allows an attacker to insert file from outside of server. These attacks have a harmful impact because the file is inserted in the form of shell [2]. By utilizing this Shell a hacker can gain access to the system that perform various activities such as viewing directory, change the page (defacing) or steal sensitive information from existing files in the system. Here's an example of writing in the RFI website:

- http://deddie.com/page.php?page=http://site.attacker/kode.tx
  t

If the site does not have good control in the the variables $ page, then the contents of “kode.txt” will be displayed in the browser. The contents of the kode.txt can be modified by an attacker to install the backdoor shell on the target system.

Typically the attacker will try to use the Remote File Inclusion first to penetrate in the website, but if the attacker knew that a form of penetration that he did fail (usually due to "allow_url_include = off" in the php.ini), will pursue an attack by a method in which the page Local File Inclusion inserted on the same server.
File Inclusion form of attack is high risk, because the attacker could gain shell access, and ultimately affect Local Exploitation in which the attacker gains full access rights to the system.

3. DESIGN

The design of this scanner security system applications are: starting the process of crawling to get a website structure, conduct experiments attacks on the pages that have security holes, and displays the results of security holes reports from the website.
The design of the application security scanner can be seen in figure 2.

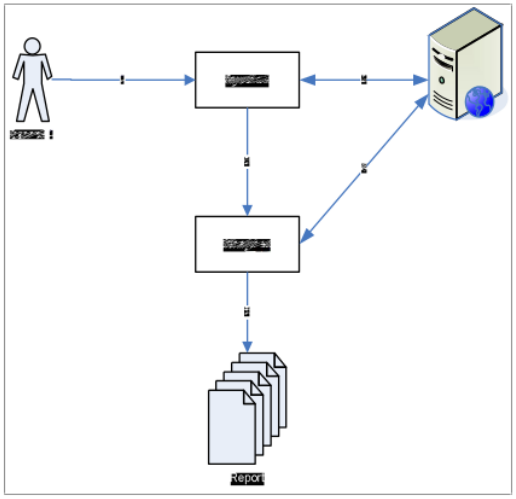


Figure 2 Design of thr Application Security Scanner

3.1 Crawling Process

Crawling process is a process where the software scans the response from the server to get the links on the website. process is performed recursively, allowing the software to index the entire link from the website. The system provides additional features that allow users to see the crawling process, log, and the structure of the crawled website, and see a link that is ignored. Crawling process can be seen in Figure 3.

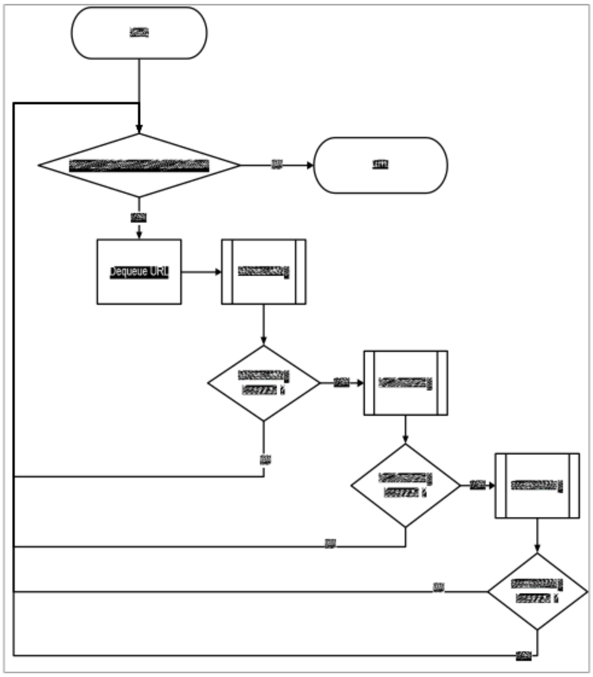


Figure 3 Crawling Process Diagram

3.2 Enqueue URL

Enqueue URL process used in the crawling process is not obtained directly insert a link to the Queue. The initial steps of this process are performed by checking whether the protocol was obtained from a valid link. This is done because there are different styles of writing on any web developer to create a website. Differences in writing style makes the software must create an additional process that links can be changed to a valid link before it is inserted into the Queue.

In the Enqueue URL process also checking whether the link is still to be obtained in the same website. It aims to restrict the software to be in the process of crawling, do not crawl links from different websites. Flowchart of this process can be seen in Figure 4.

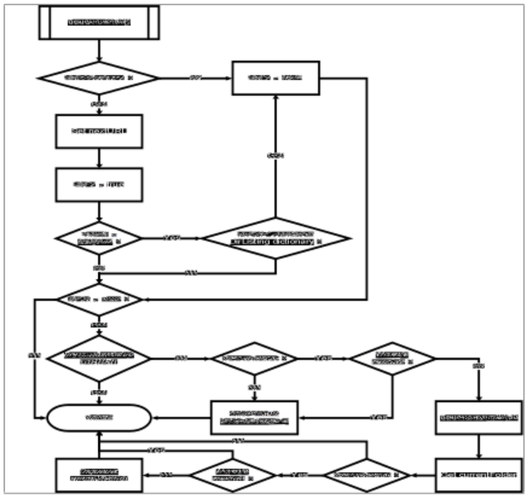


Figure 4 Flowchart of Enqueue URL Process

3.3 Attacking using LFI & RFI

Attacking Process is a process in which the experiments will be conducted attacks on the link that has been obtained from the crawling process. The result of this process is to detect whether the link has a security holes, and then will make a report about the security holes of the website.

In this process, the software will try to attack using the LFI and RFI method. Flowchart to show the LFI and RFI can be seen in Figure 5.

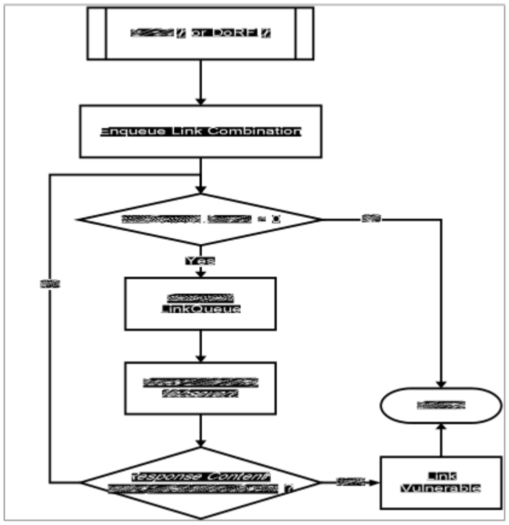
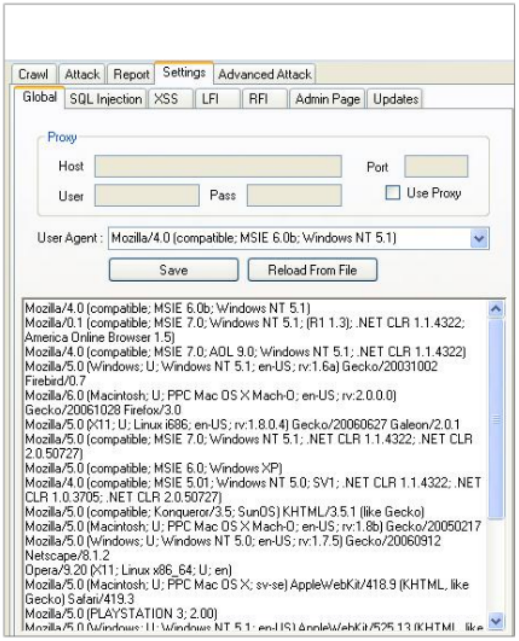


Figure 5 Flowchart of LFI and RFI method

4. IMPLEMENTATION

In this section will describe the implementation of application security scanner.

The first time when running the application, the user is asked to perform global configuration settings as shown in figure 6.



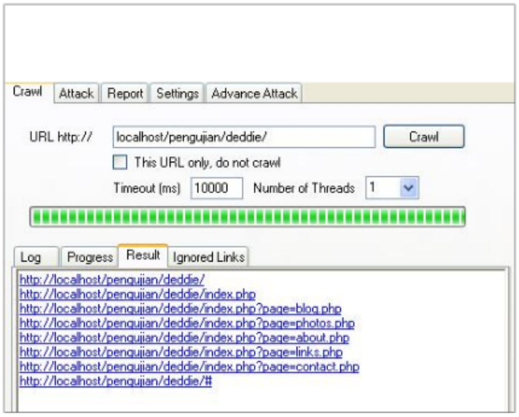
□ Use Proxy

User Agent : Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.1)

```
Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.1)
Mozilla/0.1 (compatible; MSIE 7.0; Windows NT 5.1; (R1 1.3); .NET CLR 1.1.4322;
America Online Browser 1.5)
Mozilla/4.0 (compatible; MSIE 7.0; AOL 9.0; Windows NT 5.1; .NET CLR 1.1.4322)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6a) Gecko/20031002
Firebird/0.7
Mozilla/6.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:2.0.0.0)
Gecko/20061028 Firefox/3.0
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.4) Gecko/20060627 Galeon/2.0.1
Mozilla/5.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR
2.0.50727)
Mozilla/5.0 (compatible; MSIE 6.0; Windows XP)
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0; SV1; .NET CLR 1.1.4322; .NET
CLR 1.0.3705; .NET CLR 2.0.50727)
Mozilla/5.0 (compatible; Konqueror/3.5; SunOS) KHTML/3.5.1 (like Gecko)
Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.8b) Gecko/20050217
Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.5) Gecko/20060912
Netscape/8.1.2
Opera/9.20 (X11; Linux x86_64; U; en)
Mozilla/5.0 (Macintosh; U; PPC Mac OS X; sv-se) AppleWebKit/418.9 (KHTML, like
Gecko) Safari/419.3
Mozilla/5.0 (PLAYSTATION 3; 2.00)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like
```

Figure 6 Global Configuration

The main process starts from the crawling process on the website that has been conditioned to have security holes. The first step is to get all the links that exist in the target website by crawling process, which can be seen in Figure 7.



URL http:// localhost/pengujian/deddie/

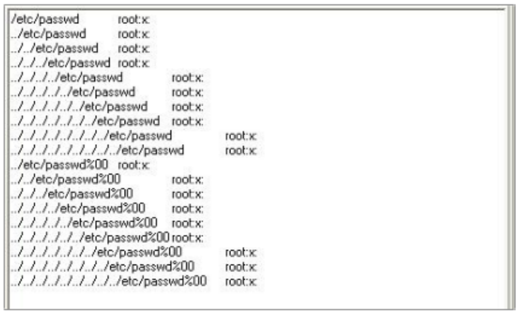
□ This URL only, do not crawl

Timeout (ms) 10000 Number of Threads 1

```
http://localhost/pengujian/deddie/
http://localhost/pengujian/deddie/index.php
http://localhost/pengujian/deddie/index.php?page=blog.php
http://localhost/pengujian/deddie/index.php?page=photos.php
http://localhost/pengujian/deddie/index.php?page=about.php
http://localhost/pengujian/deddie/index.php?page=links.php
http://localhost/pengujian/deddie/index.php?page=contact.php
http://localhost/pengujian/deddie/#
```

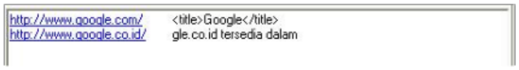
Figure 7 Crawling Process

The next process is to perform database modification of the LFI and RFI database. LFI database modifications can be seen in figure 8, while the RFI database modifications can be seen in Figure 9.



```
/etc/passwd          root:x:
../etc/passwd        root:x:
../../etc/passwd     root:x:
../../../etc/passwd  root:x:
../../../../etc/passwd           root:x:
../../../../../etc/passwd        root:x:
../../../../../../etc/passwd     root:x:
../../../../../../../etc/passwd  root:x:
../../../../../../../../etc/passwd          root:x:
../../../../../../../../../etc/passwd       root:x:
../etc/passwd%00  root:x:
../../etc/passwd%00              root:x:
../../../etc/passwd%00           root:x:
../../../../etc/passwd%00        root:x:
../../../../../etc/passwd%00     root:x:
../../../../../../etc/passwd%00 root:x:
../../../../../../../etc/passwd%00          root:x:
../../../../../../../../etc/passwd%00       root:x:
../../../../../../../../../etc/passwd%00    root:x:
```

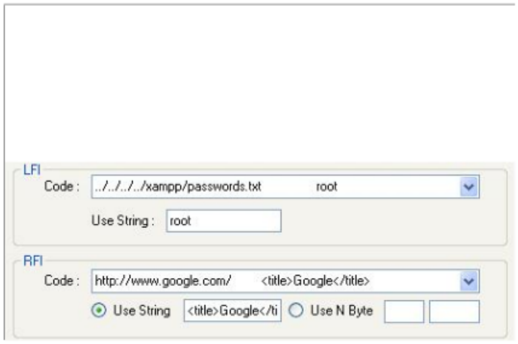
Figure 8 LFI database modifications



```
http://www.google.com/      <title>Google</title>
http://www.google.co.id/    gle.co.id tersedia dalam
```

Figure 9 RFI database modifications

In the attacking process, testing is done by changing the configuration of LFI column for testing using localhost. Then the application will try to open a file passwords.txt located in XAMPP folder by using the string root as a marker.
In the RFI testing, trying to open the address "http://google.com" using "Google <title> <title>" string, as a marker of security holes. Configuration is shown in figure 10.

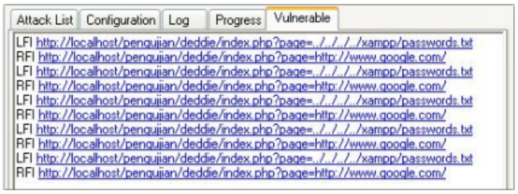


LFI
Code : ../../../../xampp/passwords.txt   root
Use String : root

RFI
Code : http://www.google.com/   <title>Google</title>
● Use String <title>Google</ti ○ Use N Byte

Figure 10 Configuration of LFI and RFI test.

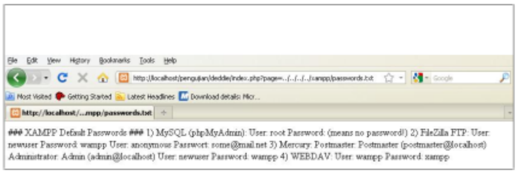
The results of the attack can be seen in Figure 11.



```
LFI http://localhost/pengujian/deddie/index.php?page=../../../../xampp/passwords.txt
RFI http://localhost/pengujian/deddie/index.php?page=http://www.google.com/
LFI http://localhost/pengujian/deddie/index.php?page=../../../../xampp/passwords.txt
RFI http://localhost/pengujian/deddie/index.php?page=http://www.google.com/
LFI http://localhost/pengujian/deddie/index.php?page=../../../../xampp/passwords.txt
RFI http://localhost/pengujian/deddie/index.php?page=http://www.google.com/
LFI http://localhost/pengujian/deddie/index.php?page=../../../../xampp/passwords.txt
RFI http://localhost/pengujian/deddie/index.php?page=http://www.google.com/
LFI http://localhost/pengujian/deddie/index.php?page=../../../../xampp/passwords.txt
RFI http://localhost/pengujian/deddie/index.php?page=http://www.google.com/
```

Figure 11 Result of Attacking Process.

In the figure 10 shows that there are security holes in the LFI and RFI. For those reasons, the testing will proceed with trying to manually request the link. In this test uses link "http://localhost/pengujian/deddie/?page=../../../../xampp/passwords.txt." to prove the security holes that exist. The results of this test can be seen in Figure 12.


File Edit View History Bookmarks Tools Help
http://localhost/pengujian/deddie/index.php?page=../../../../xampp/passwords.txt
Google
Most Visited Getting Started Latest Headlines Download details: Micr...
http://localhost/...mpp/passwords.txt

XAMPP Default Passwords ### 1) MySQL (phpMyAdmin): User: root Password: (means no password!) 2) FileZilla FTP: User: newuser Password: wampp User: anonymous Passwort: some@mail.net 3) Mercury: Postmaster: Postmaster (postmaster@localhost) Administrator: Admin (admin@localhost) User: newuser Password: wampp 4) WEBDAV: User: wampp Password: xampp

Figure 12 Result of LFI Test

RFI test by doing manually in the browser request to prove the validity of the application. Link that selected for testing was "http://localhost/pengujian/deddie/?page=http://google.com". The results of the RFI test can be seen in Figure 13.

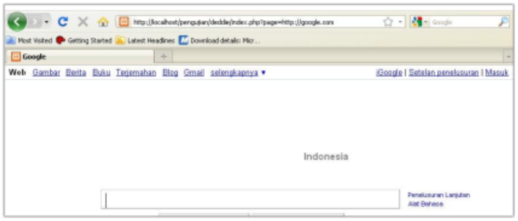

http://localhost/pengujian/deddie/index.php?page=http://google.com
Google
Most Visited Getting Started Latest Headlines Download details: Micr...
Google
Web Gambar Berita Buku Terjemahan Blog Gmail selengkapnya ▾
iGoogle | Setelan penelusuran | Masuk
Indonesia
Penelusuran Lanjutan
Alat Bahasa

Figure 13 Result of RFI Test

5. CONCLUSION

- The crawling process can affect the outcome of the report is generated; this is because the link that will be processed on the next stage (Attacking) is a link that has been obtained in the process of crawling.
- In the field testing LFI vulnerability allows an attacker to open any file by simply entering the URL path. When the website was found loopholes in this section attacker could potentially get the contents of sensitive files, which usually are targeted is a file containing the configuration of the web server.
- In testing the security gap in the field of RFI, can be seen that on a website that has this vulnerability could inject a file from another website. This would allow an attacker to inject php shell to the target website. Where the php shell, the attacker can mess up the contents of the website, it can even take over control of the target website.

6. REFERENCES

[1] Black, Paul et al. 2008. Software assurance tools: Web application security scanner functional specification version 1.0. National Institute of Standards and Technology. United States.

[2] Heydari, Mir Seyed Ali. 2008. A new source code auditing algorithm for detecting LFI and RFI in PHP programs. World Academy of Science, Engineering and Technology. Izlamic Azad University, Iran.

[3] Huang, Yao-Wen et al. 2004. Non-detrimental web application security scanning, ISSRE, pp.219-230, 15th International Symposium on Software Reliability Engineering (ISSRE'04).Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[4] Johns, Martin et al. 2008. XSSDS: Server-side detection of cross scripting attacks, 24th Annual Computer Security Applications Conference (ACSAC '08), pp. 335 - 344, IEEE Computer Society.

[5] Kals, Stefan, et al. 2006. Secubat: A web vulnerability scanner. Technical Security University of Vienna.

[6] Orloff, J. 2009. Web application security: Testing for vulnerabilities. New York: Sequoia Media Services. Inc.

[7] Stasiak, K. 2002. Web application security. Ohio: SecureState, Inc.

ICSIIT 2012-1

ORIGINALITY REPORT

3%	2%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Namibian College of Open Learning Student Paper	2%
2	Setiabudi, Djoni Haryadi, Gregorius Satia Budhi, I Wayan Jatu Purnama, and Agustinus Noertjahyana. "Data mining market basket analysis' using hybrid-dimension association rules, case study in Minimarket X", 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering, 2011. Publication	1%
3	eprints.utm.my Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		