Path Traversal - Bypass Methods April 2011

By: Pouya Daneshmand
Email: whh_iran -AT- yahoo -DOT– com
Blog: Pouya.Securitylab.ir

Path Traversal - Bypass Methods

Introduction:

Many web applications use a file to relocate downloadable files on server. for example: http://www.site.com/dl/download.php?getfile=S212.pdf

Normally, when there is a request like this, the application will load a file (for example: C:\ Inetpub\wwwroot\dl\S212.pdf) from server and show it to the user.

But Because the file name can be changed by user, If server supports special characters (../","/~) , an attacker can have unauthorized access to Sensitive data like site's configuration file and other sensitive files on server.
For example we can get source file of `download.php` with a request like this: /dl/download.php?getfile=download.php

Suppose the attacker knows the physical place of our configuration file (ex: C:\Inetpub\wwwroot \config.php) So the file can be accessed from URL below: /dl/download.php?getfile=/config.php Or maybe the attacker wants to read boot.ini: /dl/download.php?getfile=../../../../boot.ini

But Remember, if there is a protector program on the server, this aim will not be reached as easy as this!

Path Traversal - Bypass Methods

Bypass Methods:

Path traversal vulnerabilities are very common in the process of programming web applications. The parameters which can be abused or filtered/removed are obviously known, But there is a risk about filtering and that is particular methods used by attackers. Here are some methods to prevent this vulnerability to be happened, However they can be used in combination.

- Encoding Data:

for example encode the name of your file with common encoding methods:

download.php?getfile=Y29uZmlnLnBocA==

in above example the file name is base64 encoded.

- Encrypting Data:

In this method you must use encrypting algorithms:

download.php?getfile=%63%6F%6E%66%69%67%2E%70%68%70%0A%09%09%09

- Special Characters:

In some web apps, it's possible to use special chars (like ~/ ../) in file names.This can be used by attackersto changing current directiory to another one.

example: download.php?getfile =~/../ boot

- File Extensions:

Some web apps validate file's extension before letting it to be downloaded.
An attacker can bypass this using null byte! for example: "./../../../ boot.ini% 00.jpg"
This way web apps which use OS API will have this vulnerability because OS and web server have Different interpretations about null bytes. In this example ,the real file name will be cutted down using API System to "./../../../ boot.ini". You may also use %20 (ex: ../../../index.asp%20) .

Path Traversal - Bypass Methods

- Using HTTP Referrer:

In This method the attacker directly edits HTTP Referrer value. Most web servers can't validate these types of variables correctly and will trust them by default.