ISO/IEC JTC 1/SC 22/WG 23 N 0269

Possible new vulnerability, Unrestricted file upload (CBF)

Date	2010-08-31
Contributed by	*John Benito*
Original file name	
Notes	

7.nn Unrestricted File Upload [CBF]

7.nn.1 Description of application vulnerability

A first step often used to attack is to get an executable on the system to be attacked. Then the attack only needs to execute this code. Many times this first step is accomplished by unrestricted file upload. In many of these attacks, the malicious code can obtain the same privilege of access as the application, or even administrator privilege.

7.nn.2 Cross reference

CWE:

434.Unrestricted Upload of File with Dangerous Type

7.nn.3 Mechanism of failure

There are several failures associated with an uploaded file:

- Executing arbitrary code.
- Phishing page added to a website.
- Defacing a website.
- Creating a vulnerability for other attacks.
- Browsing the file system.
- Creating a denial of service.
- Uploading a malicious executable to a server, which could be executed with administrator privilege.

7.nn.4 Avoiding the vulnerability or mitigating its effects

Software developers can avoid the vulnerability or mitigate its ill effects in the following ways:

- Allow only certain file extensions, commonly known as a white-list.
- Disallow certain file extensions, commonly known as a black-list.
- Use a utility to check the type of the file.
- Check the content-type in the header information of all files that are uploaded. The purpose of the content-type field is to describe the data contained in the body completely enough that the receiving agent can pick an appropriate agent or mechanism to present the data to the user, or otherwise deal with the data in an appropriate manner.
- Use a dedicated location, which does not have execution privileges, to store and validate uploaded files, and then serve these files dynamically.
- Require a unique file extension (named by the application developer), so only the intended type of the file is used for further processing. Each upload facility of an application could handle a unique file type.
- Remove all Unicode characters and all control characters[1] from the filename and the extensions.

[1] See http://www.ascii.cl/control-characters.htm

- Set a limit for the filename length; including the file extension. In an NTFS partition, usually a limit of 255 characters, without path information will suffice.
- Set upper and lower limits on file size. Setting these limits can help in denial of service attacks.

All of the above have some short comings, for example, a GIF (.gif) file may contain a free-form comment field, and therefore a sanity check of the files contents is not always possible. An attacker can hide code in a file segment that will still be executed by the application or server. In many cases it will take a combination of the techniques from the above list to avoid this vulnerability.