URL manipulation attacks

Introduction to URLs

The URL (Uniform Resource Locator) of a web application is the vector that makes it possible to indicate the requested resource. It is a string of printableASCII characters that is divided into five parts:

- **The name of the protocol**: this is in some sorts the language used to communicate on the network. The most widely used protocol is the HTTP protocol (*HyperText Transfer Protocol*), which makes it possible to exchange web pages in HTML format. A variety of other protocols may also be used (FTP, News, Mailto, etc.)
- **ID and password**: makes it possible to specify the parameters required to access a secure server. This option is not recommended since the password circulates unscrambled in the URL
- **The name of the server**: This is the domain name of the computer hosting the requested resource. Note that it is possible to use the server's IP address.
- **The port** number: this is a number associated with a service that tells the server what type of resource is being requested. The port that is associated with the protocol by default is port number 80. When the server's web service is associated with port number 80, specification of the port number is optional.
- **The access path to the resource**: This last part tells the server where the resource is located, that is, in general, the location (directory) and the requested file name.

A URL has the following structure:

Protocol	Password (optional)	Server name	Port (optional if 80)	Path
http://	user:password@	www.commentcamarche.net	:80	/glossair/glossair.php3

The URL can make it possible to send parameters to the server by following the file name with a question mark and then data in ASCII format. A URL is then a string of characters with the following format:

http:*//en.kioskea.net/forum/?cat=1&page=2*

URL manipulation

By manipulating certain parts of a URL, a hacker can get a web server to deliver web pages he is not supposed to have access to.

On dynamic websites, parameters are mostly passed via the URL as follows:

http://target/forum/**?**cat=2
The data present in the URL are automatically created by the site and when navigating normally, a user simply clicks on the links proposed by the website. If a user manually modifies the parameter, he can try different values, for example:

http://target/forum/**?**cat=6
If the designer has not anticipated this possibility, the hacker may potentially obtain access to an area that is usually protected.

In addition, the hacker can get the site to process an unexpected case, for example:

http://target/forum/**?**cat=**********
In the above example, if the site's designer has not anticipated the case where the data is not a number, the site may enter an unexpected state and reveal information in an error message.

Trial and error

A hacker may possibly test directories and file extensions randomly in order to find important information. Here a few classic examples:

- Search for directories making it possible to administer the site:
- **http:**//target/admin/
  http://target/admin**.**cgi
- Search for a script to reveal information about the remote system:
  http://target/phpinfo**.**php3
- Search for backup copies. The *.bak* extension is generally used and is not interpreted by servers by default, which can cause a script to be displayed:
  http://target/**.**bak
- Search for hidden files in the remote system. On UNIX systems, when the site's root directory corresponds to a user's directory, the files created by the system may be accessible via the web:
- **http:**//target/**.**bash_history
  http://target/**.**htaccess

Directory traversal

So-called *directory traversal* or *path traversal* attacks involve modifying the tree structure path in the URL in order to force the server to access unauthorized parts of the site.

In a classic example, the user may be forced to gradually move back through the tree structure, particularly in the event that the resource is not accessible, for example:

http:*//target/base/test/ascii.php3*
http:*//target/base/test/*
http:*//target/base/*
On vulnerable servers, attackers can simply move back through the path with several "../" type strings:

http://target/**../../../../**directory/**file**
More advanced attacks encode certain characters:

- either in the form of URL encoding:
  http://target/**..%**2F**..%**2F**..%**2Fdirectory/**file**
- or with a Unicode notation:
  http://target/**..%**u2216**..%**u2216directory/**file**

Many dynamic sites pass the name of pages to be displayed as parameters in a form similar to the following:

http://target/cgi-bin/script**.**cgi**?**url=**index.**htm
If no verifications are carried out, a hacker may modify the URL manually in order to request access to a site resource he does not have direct access to, for example:

http://target/cgi-bin/script**.**cgi**?**url=script**.**cgi

Countermeasures

To secure a web server against URL manipulation attacks, it is necessary to keep a watch on vulnerabilities and regularly apply the patches provided by the web server's publisher.

Moreover, a detailed configuration of the web server helps keep users from surfing on pages they are not supposed to have access to. The web server should therefore be configured as follows:

- Prevent the browsing of pages located below the website's root (*chroot* mechanism);
- Disable the display of files present in a directory that does not contain an index file ("Directory Browsing");
- Delete useless directories and files (including hidden files);
- Make sure the server protects access to directories containing sensitive data;
- Delete unnecessary configuration options;

- Make sure the server accurately interprets dynamic pages, including backup files (*.bak*);
- Delete unnecessary script interpreters;
- Prevent HTTP viewing of HTTPS accessible pages.

Source: http://en.kioskea.net/contents/31-url-manipulation-attacks