

Security Tips for NYU Webmasters

This brochure outlines several essential security procedures that NYU webmasters should follow to protect the information on their website(s). If you are already an NYU webmaster, be sure to follow the instructions provided in this publication. If you have any questions about website security, please send email to security@nyu.edu.

If you are not already a webmaster, but are interested in having a website on NYU Web, Information Technology Services offers this opportunity to everyone affiliated with NYU:

- Any student, faculty, staff member, or administrator with access to NYUHome can create a personal page on the NYUHome (homepages.nyu.edu) server.
- NYU professors can use NYU Web as a tool for the classes they teach.
- Student organizations that are registered with the Office of Student Activities can set up web pages for their organizations.
- NYU schools, departments, programs and offices are encouraged to create and maintain their own pages on NYU Web, and any affiliated faculty member, staff member or student can be approved as the webmaster for a particular official NYU website.
- Graduate students and faculty can create websites on NYU Web to showcase their thesis projects or to present their large-scale personal projects.

For more information about getting started as a webmaster at NYU, please send email to webteam@nyu.edu.

Contents

Access Restriction	2-5
UNIX Permissions	5-7
SSL Information	7-8
More Information	8



A publication of...

NEW YORK UNIVERSITY
Information Technology Services

© 2004 - NYU. All Rights Reserved. 4-04. ITS Pub.# 2004-11.
<http://www.nyu.edu/its/>. Comments? Write to its.pubs@nyu.edu.

Access Restriction

There are three ways to restrict access on the www.nyu.edu server. You cannot combine these methods; you must pick one that best suits your needs. NOTE: You cannot restrict access for sites on the homepages.nyu.edu server.

1. Deny or allow certain domains. This is useful for internal documents. For instance, we could deny access to anybody not within NYU or, say, allow access for only people at Information Technology Services (ITS).

All the information that controls this feature is placed in a file called .htaccess. Everytime the server is asked for a document it looks for a .htaccess-file in the same directory as the file. If this file exists, the restrictions described in the file are followed before the document is sent to the requesting user. So, if you want to restrict access to your directory called stats (for example), you would place the .htaccess file in your stats directory. Remember, this affects the entire directory, including sub-directories. Below are some sample .htaccess files. Make sure there are NO SPACES between the comma after the word deny and the word allow.

Allow only people at NYU.EDU:

```
<LIMIT GET>
order deny,allow
deny from all
allow from .nyu.edu
</LIMIT>
```

Allow only people from ITS:

```
<LIMIT GET>
order deny,allow
deny from all
allow from .its.nyu.edu
</LIMIT>
```

Allow everybody BUT x.nyu.edu and y.nyu.edu:

```
<LIMIT GET>
order deny,allow
deny from x.nyu.edu, y.nyu.edu
allow from all
</LIMIT>
```

2. Another way to restrict access is to institute a password scheme. Again, access is protected by directory, so everything in that directory will be protected. Using this method, a person will be prompted for a username and password. *(If the person does not have an authentication-capable browser, they will not receive this dialogue box, nor will they receive the file.)* Keep in mind that this username and password combination is not the same one the person logs in with—you will assign the username and password yourself.

The file `.htaccess` will again be used. This file should be in the directory which contains the documents to which you wish to restrict access. The contents of this file specify the name of the password file that you will create.

```
AuthUserFile /www/sites/nyu.edu/htdocs/full_pathname/.htpasswd
AuthName "Put Your Description Here"
AuthType Basic
<LIMIT GET>
require valid-user
</LIMIT>
```

NOTE:

- * If you have more than one word in the `AuthName` field, you must surround your text with quotation marks.
- * If there are extra spaces after `AuthUserFile`, the `.htaccess` restriction will not work.

The `AuthUserFile` is the absolute path to the directory where you will store the password file. Always begin with `"/www/sites/nyu.edu/htdocs/"`. That is the root of `www`. For example, if your website is at *www.nyu.edu/classes/english*, the pathname would be `/www/sites/nyu.edu/htdocs/classes/english`—and that should be where your `.htaccess` and `.htpasswd` files reside. If you are not sure what the full pathname to your directory is, SSH to that directory and type `pwd` from the command prompt, which will display the path to that directory.

The `AuthName` tells the browser to include a short description in its prompt when it asks for a password. You can put just about anything on this line, but ideally it should indicate something about the directory the user is about to log into. For example, if in the `AuthName` line you've put "English Course Information", that would appear in the dialog box the user sees when they get to the restricted page.

The file `.htpasswd` contains the passwords of the users. To create the `.htpasswd` file, log in through your `i4` account using SSH. Change directory to the directory you want to restrict access to, and type:

```
htpasswd -c .htpasswd someuser
```

for the first user (where `someuser` is the username). You will then be prompted twice for the user's password. The `-c` option causes the `.htpasswd` file to be created. For each additional user type:

```
htpasswd .htpasswd someuser
```

To delete a user from your password file, use a text editor to edit the password file, and delete the line that begins with the user's name.

NOTE:

There is no correspondence between the usernames and passwords used for accounts on the www server and usernames and passwords in any specific .htpasswd file. A user doesn't need to have an account on this system in order to be validated for access to files protected by HTTP-based authentication.

3. The last option is to base the person's username and password on the NetID they use for their DIAL and NYUHome account. We have implemented a "single sign-on" environment, where people use just one username (their NetID) and one single password in order to get into their accounts and to login to all sorts of other resources here at NYU. This single sign-on method can be used on www.nyu.edu as well. The advantage is two-fold: people don't have to remember another username or password, and after it's set up, you don't have to do anything! Of course, all the people you want to have access to the directory must have an NYU NetID.

The setup is much like the second option, except you don't need a second file for the password setup. Create a file called .htaccess in the directory which you want to password-protect. It must have the following two lines:

```
AuthType KerberosV5
AuthName "Put Your Description Here"
```

If you want specific people to be able to access the page, you need to list their NetIDs. For example:

```
AuthType KerberosV5
AuthName "Put Your Description Here"
require user netid1
require user netid2
require user netid3
```

(Replace the NetIDs in italics with the ones you wish to include.)

If you want anyone with an active NYU NetID/password to be able to access your site, then the .htaccess file should look like this:

```
AuthType KerberosV5
AuthName "Put Your Description Here"
require valid-user
```

You do not need to set up passwords because the server has access to a database which has all the NetIDs and appropriate password for each. This file must be saved as .htaccess (make sure it's in the directory you want to restrict access to). If you wish to test the site restriction, you can go to the URL of your

site and enter in your NetID/password combination. Remember, once you've logged in successfully, you'll need to quit and restart your browser in order to test again!

UNIX Permissions

UNIX allows you to designate, on a file by file basis, who has permission to read it and write it. This is known as file permissions. When you upload a file, you become the owner of that file and it is assigned to the group you are in. But, unless you say that other group members have permission to write to the file, they cannot make modifications. Here is a typical file, and how to read the UNIX permissions:

```
-rwxr--r--  username      groupname      546 Dec 10 13:10 filename
```

[-]

This area designates what kind of entry it is. If it is a file, it will simply show “-”. If it is a directory, it will show “d”.

[rwx]

These first three slots are the permissions for the owner of the file. The r means the owner can read the file... the w means the owner can write to the file... and the x means the owner can execute the file.

[r--]

These second three slots are the permissions for the group. In this case, the group members can read the file but not write it or execute it. (Execution is usually only given for a script or a directory.)

[r--]

These last three slots are the permissions for the world. In order to have your pages visible by people viewing through a WWW browser, you need this set to “readable.”

[username]

This is the owner of the file. Your username will appear in this space.

[groupname]

This is the name of the group this file belongs to. Your groupname will appear in this space. Those working on sites alone will not be in a group with other people and need not worry about group permissions.

[546]

This is the file's size.

[Dec 10 13:10]

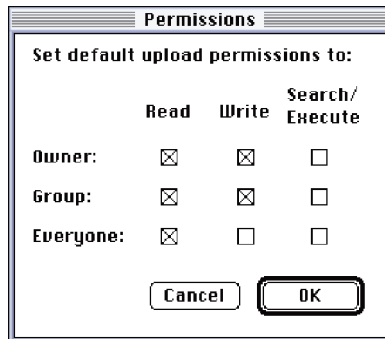
This is the last modification date (and time) of the file.

[filename]

And, of course, this is the name of the file.

When you upload a file to www, it sets the permissions by default so that the owner can read and write to the file, the group and world can only read it (-rw-r--r--). Ideally, we want the group to be able to write to it as well (-rw-rw-r--). There are a few ways to handle this.

1. If you haven't uploaded the files yet, you can tell certain FTP programs how you want the files uploaded. Log in to your account via the FTP program to the Remote menu and choose Set Upload Permissions.... Toggle the following buttons:



Now, all files during that FTP session will be uploaded with the correct permissions.

2. If the files are already on the server, you will need to SSH into your account and change them. Once you have logged in, cd to switch to your web directory. For many of you, this command might be cd web. To list the contents of the directory, type ls. You can modify ls with the following:

```
-l Long format where you can see file permissions.  
-a This will make files preceded by dots visible.  
-g This shows the group name.
```

So, you could type ls -lag and incorporate all those features into one. Remember, if you want to see the contents one page at a time, pipe the listing to the program more by typing ls -lag |more.

Go into the directory where you want to change the permissions. If you want to change everything except for the directories, issue the following command:

```
chmod 664 *.*
```

chmod means "change mode"... 664 is a number combination that will set the permissions to what we want... and *.* means all files that contain a period.

So, this will omit directories. For directories, you should issue the following command:

```
chmod 775 directoryname
```

If you want a directory or file to be unreadable, unwritable, and unexecutable by anyone but the owner, issue one of the following commands:

```
chmod 700 directoryname OR chmod 700 filename
```

Remember, you cannot be inside of the directory for which you are trying to change the permissions.

In case you are interested, the following chart shows how we get the numbers 664, 700, and 775. You just add up the numbers of the settings you want:

```
0400 Allow read by owner.
0200 Allow write by owner.
0100 Allow execute (search in directory) by owner.
0700 Allow read, write, and execute search) by owner.
0040 Allow read by group.
0020 Allow write by group.
0010 Allow execute (search in directory) by group.
0070 Allow read, write, and execute (search) by group.
0004 Allow read by others.
0002 Allow write by others.
0001 Allow execute (search in directory) by others.
0007 Allow read, write, and execute (search) by others.
```

So, 664 is 0400 + 0200 + 0040 + 0020 + 0004.

Secure Sockets Layer (SSL)

How to ensure that others can't snoop on the information individuals send through your website (for example, an ID number)

If you've done any shopping on the Web, you have probably encountered an order form on what is labeled a "secure server." "Secure" in the context of such pages has come to mean "SSL," for Secure Sockets Layer. www.nyu.edu is a secure server.

When you make a regular Web request using an http URL, traffic is sent over the network in an unencrypted state. Usually, this is nothing to worry about. If your page is publicly available, it makes little sense to encrypt it for transfer over the network; everyone can already see it!

Sometimes, however, you need to collect from or send to users sensitive data—credit card information, for example, or, more relevant in an educational environment, grades. In this case, sending the information over the network in an

unencrypted form permits snooping, that is, the act of “spying” on network traffic as it passes from point A to point B. (Remember that when you download a page from, say, Amazon.com, that page passes through possibly many other networks before reaching your computer.)

SSL adds a cryptographic layer to the standard TCP/IP protocol suite. Information is encrypted as it leaves your computer. Anyone snooping the data while it’s on the network will see only random characters, not the information as it was originally formatted.

Once the traffic reaches its final destination, the destination computer decrypts the garbled data, returning it to its original state. It then forwards the decrypted data to the recipient. To both the sender and the recipient of the data, the encryption/decryption process is transparent.

Activating SSL on www.nyu.edu requires changing any intra-site http links into https links. If you’re using relative links in your pages, then SSL-enabling your entire site might mean having to change only the entrance URL from http to https. (Experienced webmasters may know that http-style URLs map to port 80 on the server machine; https-style URLs map to port 443.) If you want to selectively activate SSL within your site, you will need to use full http or https URLs in your pages. (Don’t forget to test the links.) Note: if your site contains forms, be sure to use https for the submit link, or the information will be sent in the clear. SSL-capable browsers that follow your https links will use the encryption layer SSL provides.

For further help and information, visit:

<http://www.nyu.edu/its/security/>

Call the ITS Client Services Center at:

1-212-998-3333

Contact the NYU Webmaster at:

webteam@nyu.edu

Or send e-mail to the ITS Network Security Group at:

security@nyu.edu