OIT Web Hosting Best Practices

Revised February 15, 2013

Use these guidelines to make sure your website is successful and secure; now and in the future.

1. Do not use personal netIDs for websites

(Departmental cPanel and dotNET only)

Departmental cPanel sites are installed using a specific netID. Only that netID can create and/or edit files in the site. While dotNET accounts and sites do not use a netID as such, they do link to departmental file server space on \\files\dept. The subdirectories used are under the departmental space and sometimes in a subdirectory tree identified by netID. In both cases personal netIDs should NOT be used. There are two primary reasons: (1) if the person leaves the University or changes their role at the University, it might be very difficult to move the website out of their personal space; and (2) if help is needed for website programming or troubleshooting, it might be necessary to give the account password to someone else, which is troublesome and contrary to OIT recommendations if it is also a personal netID. Departmental/organizational netIDs can be created by filling out the form at:: https://oitforms.princeton.edu/HD/newadmacct.

2. Use your "PROD" and "DEV" server environments to fullest advantage

(Departmental cPanel and dotNET only)

With both Departmental cPanel and dotNET services you get accounts on both "production" and "development" servers. Professional developers have long known of the wisdom of being able to have one site that is actively in use while having another nearly identical development site on which to test changes and new versions of software before making it available to the public. Your "DEV" site can also be a test bed for new versions of your website and interim security scans.

3. Restrict file and directory permissions

Files and directories can be secured by restricting read and write permissions to the netID of the site. There is never a reason for general readability. In general, for cPanel use the following permissions: 755 for directories, and 644 for all other files, except for CGI executables, which get permission 701. For dotNET accounts, file server permissions on the account directory should be limited to the specific departmental or personal netIDs for those who will create and edit the files. For more information on how to do this using Windows Access Control Lists, see: http://kb.princeton.edu/9680.

4. When authenticating Princeton netIDs use CAS instead of LDAP

When your application asks for a Princeton netID and password, and then authenticates against LDAP, your application ends up with a plain-text copy of the user's password in memory, even if you are complying with OIT's rule of not asking for the password over an unencrypted connection. While we certainly hope that all Princeton web developers will do the right thing and not remember the password, it is not a secure practice, and can expose the password to malicious hackers. For this and other reasons OIT has introduced a centralized authorization service known as CAS that gets your site out of the Princeton netID authentication business. For more information, see :https://sp.princeton.edu/oit/sdp/CAS/.

5. Design your site to be hacker resistant from day one

Though no developer intentionally creates an easily hackable website, we have seen several recent examples of sites that failed to guard against SQL injection[i] or cross-site scripting[ii] attacks. While these attack methods may not be well known, they are part of the web landscape, and web application developers cannot afford to ignore them. Any manager whose staff creates web applications or who has hired consultants to do so needs to ensure that the application developers know not only what SQL injection and cross-site scripting attacks are, but also how to guard against them (see below).

6. Before "go-live," do a security scan on your website

To assist you in your development efforts OIT provides a free service that can significantly improve your website's security. You should use this tool before publicly announcing your site to the world (which, unfortunately, also makes it known to the hacker community). OIT has software called IBM AppScan that will scan your website and produce a security vulnerability report, which is extremely helpful in identifying weak spots and suggesting ways to fix them. For more information, see http://www.princeton.edu/itsecurity/services/checkup/.

7. Plan for change

If your website has successfully passed AppScan's review, you may publish it with confidence in its security. However, the reality is that the hacker community will continue to develop new exploits for attacking your website, and you must remain vigilant. If you are running open source software such as Drupal or WordPress, you absolutely need to keep current with your security upgrades or you are putting out the welcome mat for hackers. If you are a manager whose website has been developed with open source software by a consultant, you will need to keep your consultant on retainer or find another consultant to keep your site secure. You may want to check with your accounting department to make sure your operating budget includes funds to remediate your web applications if needed. Website security remediation will not be provided by OIT unless OIT created the web application that was compromised.

[i] SQL injection attacks are possible when a user of the website can invoke a web page and supply their own SQL commands to be executed. For more details, see http://en.wikipedia.org/wiki/SQL_injection.

[ii] Cross-site scripting attacks are a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users. For more information, see http://en.wikipedia.org/wiki/Cross-site_scripting.