nLightened Thoughts

June 2013

William Blinn Communications
Worthington, Ohio 43085
www.n-lighten.us • 614/859.9359

Security is Important for All Websites. Here's Why.

Your website doesn't sell any product or service directly. You don't accept online payments. No credit card information is stored on your site. You don't even ask people to register their name or e-mail address. So security is no big deal, right?

Wrong.

In fact, your site might be an ideal target for cyber-crooks. Not for the valuable data they can find and carry away (we've already said that there isn't any). Instead, the crooks will find security holes in your site and exploit them to plant malware on visitors' computers. Or they'll surreptitiously create a directory or a sub-site on your server and use it for their illegal acts.

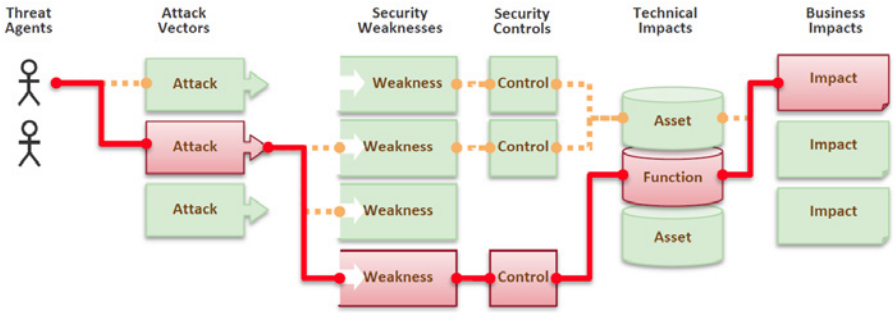
This will not be popular with clients or prospective clients.

The Open Web Application Security Project (OWASP) Top Ten Project lists the 10 most serious threats each year and each new year's report bears a distressing similarity to the previous year's report. Actually, this is to be expected because the report doesn't list specific exploits that take advantage of a particular piece of software or hardware, but instead focuses on types of errors that can be exploited.

OWASP is a non-profit organization that was founded in 2001 and counts 36,000 participants worldwide. OWASP's mission involves explaining online security threats, particularly those that are Web based, and providing crowd-based guidance for mitigating the problems.

Most of the top ten threats have been the top ten threats for the last ten years and there's no likelihood that they'll fade away anytime soon even though they've been well identified.

Programmers, being both human and creatures of habit, tend to make the same kinds of mistakes year after year. Hence the similarities from year to year in OWASP's report.



First released in 2003, the goal of the Top 10 project is to raise awareness about application security by identifying some of the most critical risks facing organizations. It's really only the beginning, though. OWASP says: "There are hundreds of issues that could affect the overall security of a Web application." In fact, neither the overall Internet nor the area known as the Web was designed with security in mind. Initially, these were technologies designed for sharing information and there was no expectation that they would be used for commerce.

Whatever security measures exist have been bolted on later as an afterthought. This alone explains why so many problems exist. The best solution would be simply to start over and design a secure network from the ground up. Clearly that's not going to happen anytime soon, if ever, so the second best solution is for developers to place security ahead of functionality. Unfortunately, because functionality that provides known benefits has more intrinsic appeal that protecting against unknown threats, that rarely happens.

So realistically, all we can hope for is an environment in which developers carefully assess the risk of each type of technology they use and actively build in protections that mitigate the threats.

For starters, any application or website should be reviewed to ensure that the OWASP Top 10 threats have been resolved. These aren't obscure or unknown threats and most of them have clear strategies for remediation.

Why So Many Problems?

It's difficult to understand why some of the attack vectors continue to be in the top 10 year after year.

Take, for example, the number 1 threat, "Injection". An injection occurs when untrusted data is sent to a command interpreter. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.

nLightened Thoughts (ISSN 1543-1533) is for clients and friends of n-Lighten.us, 179 Caren Avenue, Worthington, Ohio 43085. Phone 614.859.9359. On the Web at www.n-Lighten.us.

Example: A website form asks for a name, a phone number, or an e-mail address and depends solely on browser-based validation. When the value entered by the user reaches the Web server, it is passed to the database server without further testing. In this scenario, a crook can include a bit of punctuation and a command that would cause the server to return a dump of the database.

The solution is simple: Regardless of any validation that's performed in the browser, all data that arrives at the server must be considered suspect. The developer simply creates what's called a white list of acceptable characters. Only those characters (typically numbers, upper- and lower-case letters, and specific punctuation characters) are accepted. Everything else is removed before the value is passed on to the database.

The other primary vulnerabilities range from broken authentication and cross-site scripting to security misconfiguration and failure to install security updates.

Many of the threats are difficult for people who are not developers to understand without complex and ponderous explanations, but security updates are both critically needed and easy to comprehend.

Simple Script control panel for a BlueHost user.

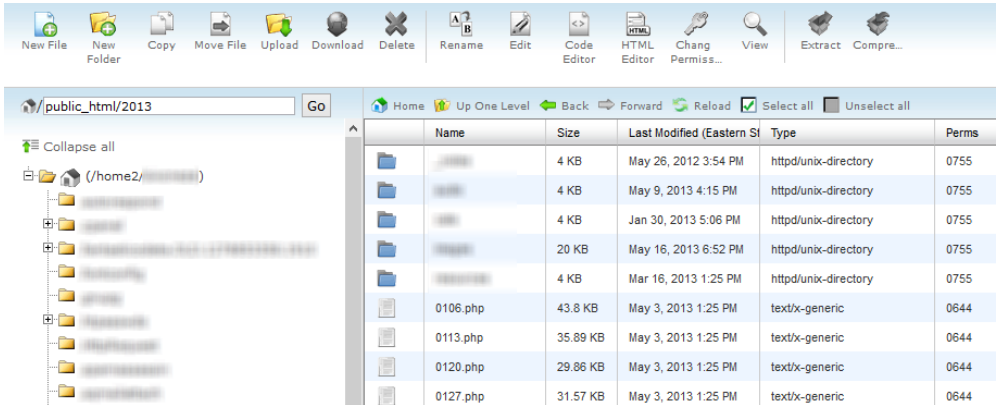
Most website hosting companies offer various add-ons. BlueHost, for example, offers a service called Simple Scripts that provides several dozen applications ranging from content management systems and guest books to forums, surveys, and full website development tools. Each of these applications is routinely updated.

Simply put: No application is ever perfect.

Website Attacks

One of the easiest but most critical safeguards is making sure that anything installed on your Web server is up to date.

BlueHost, the service most of my clients use, does an excellent job of keeping the base software patched and updated. As with virtually all site hosting services, BlueHost runs open-source software that is frequently updated.



It's easy to be intimidated by Linux permissions, but it's equally easy to learn how the numbers are composed and what they mean. The most common permission settings are 755 for directories and 644 for files.

It's good that the code is open-source because the good guys can look for weaknesses that might be exploited and fix them. It's also bad that the code is open-source because the bad guys can look for weaknesses that might be exploited and use them. That's why updates are so important.

Few, if any, hosting operations will update any applications that you have installed. You can, for example, install WordPress manually or via the Simple Scripts interface, but then it's up to you to check for updates regularly and install them. The hosting companies won't do this because updates can break existing installations if those installations have been improperly modified by the user.

Some content management systems such as WordPress accept plug-ins that can improve security, but these need to be updated frequently, too. Or you might consider SiteLock (www.sitelock.com), a third-party service that's offered by many hosting companies. SiteLock monitors your site daily to detect malware, identify vulnerabilities, and scan for any virus code that may have been planted on your site.

777 is Not a Lucky Number

Most websites reside on Linux severs and Linux controls folder and file access by assigning what are called read, write, and execute permissions for each of 3 classes of users: 777 means everyone can do anything to the file.

The classes are used to identify the file or folder owner, the group that the file owner belongs to, and everyone else (aka "the world").

Permissions are stated numerically: 4 means the user can read the file or folder, 2 allows writing, 1 indicates execute permission, and 0 (as you probably expect) means no privileges.

Each of the 3 user types is represented by a single number: 751, for example, means that the file owner can read, write, and execute the file (7=4+2+1), the group can read and execute the file but not modify it (5=4+1), and all other users may only execute the file but not read or write it.

Inexperienced website developers sometime try to install a Perl script (Perl is a scripting language that's commonly used on websites), finds that the script won't run because of a problem with permissions and sets the file and the directory that contains it to 777. Bad move! Now anyone who has access to the site can modify the file.

In most cases, folders should be set to 755 and files to 644.

Acceptable Risks

Some risks are acceptable. We live in a world of acceptable risks.

When we drive across town, we know that we could be run over by a large truck or hit by a falling meteor. Both of these are unlikely, so most of us accept the risk. With websites, there are acceptable risks, too, but any exploit that has been clearly defined—and particularly those in the OWASP Top 10—must be eliminated.

The threats are real. Don't be sorry. Ω