Configure a Safe Environment for PHP Web Apps

Zend Core for i5/OS provides options to safeguard your system

by Alan Seiden

WITH ITS SUPPORT OF THE POPULAR web programming language PHP, the System i runs a large variety of PHP-based software for the web. Given the Internet's open nature, prudent system administrators who deploy web applications in IBM's Zend Core for i5/OS environment will want to ensure security. What precautions are needed?

Although System i's architecture automatically protects against buffer overruns, viruses, and worms, and Zend Core PHP provides additional safeguards beyond those that exist in generic PHP, you should add safeguards against other dangers, such as

- propagating viruses to browsers (even if the site itself is immune)
- password "sniffing" (stealing)
- unauthorized running of applications
- disclosure or alteration of private data

Some of these safeguards require specific PHP programming techniques, which I'll discuss in a future article. Here, I discuss how you can obtain broad protection within the PHP environment itself.

Note: Web security is a rapidly evolving field. All the recommendations I discuss here are believed to be accurate as of Zend Core for i5/OS version 2.0.1 (i.e., PHP 5.2.1).

The First Line of Defense

No matter what applications you run, a secure runtime environment reduces your risk of known and unknown problems. PHP security expert Chris Snyder, co-author with Michael Southwell of *Pro PHP Security* (Apress 2005), recommends multiple levels of protection — so-called "defense in depth" — because "you don't know what will go wrong." Snyder, a web developer at the Fund for the City of New York, considers a web application to be only as safe as the environment in which it runs.

The PHP execution environment is the outer perimeter of application security; you want to stop attacks here if at all possible. This article deals with that outer ring of defense, including elements such as PHP and application patch maintenance, encryption, directory structures, configuration files, and the regular updating of PHP.

Keep PHP and Apps Up to Date

Each new release of PHP improves security by eliminating vulnerabilities reported by the PHP community. Between releases, Zend issues "hot fixes" — temporary patches that correct serious bugs that could compromise security until the next release becomes available.

Zend's Shlomo Vanunu recommends that administrators keep current by configuring automatic Zend Network updates, which will apply patches and release updates as needed to maintain security. Vanunu, a senior consultant in Zend's lab in Ramat-Gan, Israel, and a team leader of Zend's i5 Global Services department, notes that IBM has arranged for all System i customers to get these updates through a free subscription to Zend Network Silver Support.

Here are the steps to configure automatic updates:

1. Register at Zend Network (*zend.com/network*). You may have done this already if you downloaded Zend Core from the web.
2. From a System i command line, type in the following:

```
GO ZENDCORE/ZCMENU
```

3. Choose menu option 2 — Update via Zend Network.
4. From the Update menu, choose to update immediately or on a schedule.

Besides PHP's own updates, popular PHP-based web applications issue their own patches and regularly release upgrades. According to Chris Snyder, even the best projects end up with some bugs that can be dangerous. So it behooves everyone to stay informed about vulnerabilities and corresponding updates using these methods: