КЛАССИФИКАЦИЯ УЯЗВИМОСТЕЙ СЦЕНАРИЕВ НА ЯЗЫКЕ PHP

Д.А. Дука, А.Н. Мацкевич

Современные версии PHP позволяет создавать изображения, PDF-файлы, флэш-ролики, в него включена поддержка большого числа современных баз данных, встроены функции для работы с текстовыми данными любых форматов, включая XML, и функции для работы с файловой системой. Он поддерживает взаимодействие с различными сервисами посредством соответствующих протоколов, таких как протокол управления доступом к директориям LDAP, протокол работы с сетевым оборудованием SNMP, протоколы передачи сообщений IMAP, NNTP и POP3, протокол передачи гипертекста HTTP и т.д. Так же включена поддержка объектов Java и возможность их использования в качестве объектов PHP.

Языки описания сценариев, такие как Perl, Python, Rexx, Tcl, PHP, ASP и языки оболочек UNIX, предполагают стиль программирования, весьма отличный от характерного для языков системного уровня. Они предназначаются не для написания приложения с нуля, а для комбинирования компонентов, набор которых создается заранее при помощи других языков. С появлением этих языков программирования возникли и новые возможности по взлому систем, в которых они используются.

Предлагается следующая классификация уязвимостей сценариев на языке PHP:

– неправильная установка интерпретатора PHP;

– неправильная настройка интерпретатора PHP (опции Register_globals, Magic Quotes и т.д.);

– некорректное использование конструкций языка PHP (Fopen, Include, Require и т.д);

– переполнение памяти в следствии отсутствия деструкторов в классах;

– неправильная работа с сессиями и COOKIE;

– вывод критичных (с точки зрения безопасности) сообщений об ошибках;

– компрометация исходного кода при выходе из строя WEB сервера;

– выполнение системных вызовов из PHP сценариев;

– использования сценариев на PHP в качестве прокси-сервера;

– некачественная фильтрация входных данных пришедших от пользователей;

– загрузка вредоносного кода под видом картинок и другой информации.

Большинство успешных атак основывается на коде, написанном без учета соответствующих требований безопасности. В частности открытые проекты постоянно сообщают о найденных новых ошибках в их коде.

Поэтому необходимо учитывать особенности языков программирования с точки зрения классификации уязвимостей и мер защиты информации.

Литература

1. Котеров Д.В., Костарев А.Ф. PHP 5. Наиболее полное руководство. 2007.
2. Фленов М.Е. PHP глазами хакера. СПб., 2005.