|  |
| --- |
| **漏洞基本信息** |
| **漏洞名称** | Subrion CMS-database-php-SQL注入 |
| **受影响实体版本号** | Subrion CMS 4.2.1 |
| **漏洞类型** | SQL注入 |
| **危害等级** |  |
| **漏洞简介** | Subrion 是一个\*\*内容管理系统\*\*(CMS)，它允许您为任何目的构建网站，是一个功能强大的 Web 应用程序，需要一个带有 PHP / MySQL 的服务器才能运行。在 Subrion CMS 4.2.1 的项目文件 admin/database.php 中存在SQL注入漏洞，攻击者利用该漏洞通过注入SQL语句可以实现脱裤甚至GetShell。 |
|  |
| **漏洞验证** |
| **漏洞定位** | http://test.com |
| **漏洞触发条件** | 管理员身份登录后台 |
| **漏洞验证过程** | 后台调试代码，定位到 includes/classes/ia.base.controller.admin.php 文件，代码 127 行根据 action 参数选择 switch 语句的执行代码：// 第 127 行代码 if (iaView::REQUEST\_HTML == $iaView->getRequestType()) { switch ($iaView->get('action')) { # 0:sql case iaCore::ACTION\_READ: $this->\_indexPage($iaView); break;继续追踪，跳转至 admin/database.php 文件代码 221 行的 \_queryPage() 函数，其中 224 行调用 $this->\_iaDb->query() 函数执行 SQL 语句，针对要执行的 SQL 语句值只使用 trim() 函数进行简单过滤， 没有严格地过滤数据库相关字符，导致SQL注入漏洞：// 第 221 行代码private function \_queryPage(&$iaView) { if (isset($\_SESSION['queries'])) { $iaView->assign('history', $\_SESSION['queries']); } if (isset($\_POST['exec\_query'])) { iaUtil::loadUTF8Functions('ascii', 'validation', 'bad', 'utf8\_to\_ascii'); $sql = $\_POST['query']; $outerData = ''; utf8\_is\_valid($sql) || $sql = utf8\_bad\_replace($sql); $queries = (false === strpos($sql, ';' . PHP\_EOL)) ? [$sql] : explode(";\r\n", $sql); foreach ($queries as $key => $sqlQuery) { $sql = trim(str\_replace('{prefix}', $this->\_iaDb->prefix, $sqlQuery)); $this->\_iaCore->startHook('phpAdminBeforeRunSqlQuery', ['query' => $sql]); // 244 行 $result = $this->\_iaDb->query($sql); |