

Gefährlich SQL-Injection-Schwachstelle in WordPress

Ende 2022 haben Forscher von Patchstack insgesamt drei kritische Sicherheitslecks in Learnpress entdeckt. Bei zweien handelt es sich um klassische SQL-Injection-Attacken, wobei eine der beiden eine Contributor-Rolle im CMS seitens des Angreifers erfordert. Die andere Schwachstelle kann ein Angreifer ohne jegliche Authentifizierung ausnutzen, sie ist demnach deutlich kritischer.

Eine SQL-Injection entsteht durch einen Fehler im Programmcode, der auf die SQL-Datenbank zugreift. Ein lokaler oder entfernter Angreifer kann dadurch SQL-Befehle ausführen. Je nach Applikation und Datenbankkonfiguration kann er so die Datenbank auslesen, Daten modifizieren oder Einträge aus der Datenbank löschen. Die konkrete SQL-Injection-Attacke im Learnpress-Plugin hat zur Folge, dass ein entfernter Angreifer beispielsweise neue Konten anlegen kann. Unter anderem kann er so auch einen Administrator-Zugang erstellen und damit weit-reichende Kontrolle erlangen.

Der Programmierfehler befindet sich in der execute-Funktion der Datei `inc/databases/class-lp-db.php`. Die Funktion verarbeitet die Variable `$filter`, die SQL-Filteranweisungen enthalten kann.

Diese Variable ist ein Objekt der Klasse `LP_Filter`, die die Variable `order_by` und `order` enthält. Deren Werte kann ein Angreifer modifizieren.

Die execute-Funktion verwendet diese Strings zum Zusammenbauen der Zeichenkette `$ORDER_BY`. `$ORDER_BY` wird dabei direkt eingefügt, ohne den String zuvor auf problematische Zeichen zu überprüfen.

Die Lösung dieses Problems besteht nun darin, `$ORDER_BY` zu

filtern. Das übernimmt die Funktion `sanitize_sql_orderby` (Listing 1). Die PHP-Funktion `preg_match` überprüft, ob der übergebene Text einem Muster entspricht, das per regulärem Ausdruck definiert ist.

Mithilfe dieses Patches wird die SQL-Injection-Attacke vereitelt.

Weitere Infos und
interessante Links

www.lm-online.de/qr/47375

<https://patchstack.com/articles/multiple-critical-vulnerabilities-fixed-in-learnpress-plugin-version/>

Listing 1: `sanitize_sql_orderby`

```
function sanitize_sql_orderby( $orderby ) {  
    if ( preg_match(  
        '/^\s*(([a-z0-9_]+|`[a-z0-9_]+`)(\s+(ASC|DESC)))?  
        \s*(,\s*(?=[a-z0-9_`])|$\))+$/i', $orderby ) || preg_match(  
        '/^\s*RAND\  
        (\s*)\s*$/i', $orderby ) ) {  
        return $orderby;  
    }  
    return false;  
}
```