10. Januar 2018

Nadja Stadelmann, Tatyana Merlo, Natalie Stalder

TBZ

Analyse Schlosslauf

M183 – Applikationssicherheit implementieren

Inhaltsverzeichnis

[Analyse Schlosslauf 1](#_Toc501521471)

[Einführung und Überblick 1](#_Toc501521472)

[Schlosslauf 1](#_Toc501521473)

[Softwareanalyse und gefundene Schwachstellen 1](#_Toc501521474)

[Sicherheitsrisiken 1](#_Toc501521475)

[Sicherheitsverbesserungen 1](#_Toc501521476)

[Resultate 1](#_Toc501521477)

[Quellen 1](#_Toc501521478)

# Analyse Schlosslauf

# Einführung und Überblick

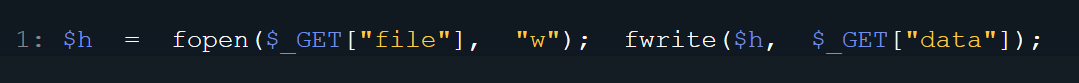
# Schlosslauf

# Softwareanalyse und gefundene Schwachstellen

## File Manipulation

Ein Angreifer kann durch Eingabefelder versuchen ausführbaren Code einzubauen, der vielleicht in einem PHP-File gespeichert wird und beim potentiellen Ausführen Schaden anrichtet.

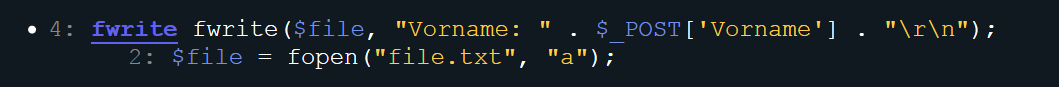
Ein Beispiel dafür ist:



### Vorhandene Probleme

Bei der Untersuchung mittels RIPS sind 14 Meldungen zu File Manipulation aufgetreten - alle im File save.php. Diese Fehlermeldungen rühren daher, dass in der Applikation Schlosslauf die Usereingaben ohne Überprüfung ins File geschrieben werden.

Ein Beispiel:



### Weiteres Vorgehen

Bei der Applikation ist deshalb sehr wichtig, dass man deshalb immer die Usereingaben auf allfällige Attacken untersucht (z.B. mit htmlspecialchars) und falls die einzugebenden Werte beschränkt sind, sogar eine Whitelist mit den zugelassenen Werten erstellt, so dass man den Userinput jeweils nur mit der Whitelist vergleichen muss.

## Cross Site Scripting

Ein Angreifer kann auf der Webseite ein JavaScript oder HTML Code ausführen lassen, wenn die Usereingaben nicht alle vor dem weiteren Verarbeitungsprozess validiert werden. Ist dies nicht der Fall und wird Userinput in den HTML Code eingebettet, kann der Angreifer durch seinen Code Scripts ausführen lassen, die ihm z.B. die Cookie-Daten oder die Session-ID verraten.

Ein Beispiel dafür ist:



### Vorhandene Probleme

Während der Untersuchung durch RIPS wurden 5 Fälle von Cross Site Scripting gemeldet. Jedoch waren dies eher Fehlmeldungen, da in der Applikation Schlosslauf zwar echo-Befehle ausgeführt werden, diese jedoch keine User-Manipulierten Daten beinhalten. Deshalb ist Cross Site Scripting nicht wirklich ein Problem in dieser Applikation.

save.php, index.php , login.php

Ein Beispiel:



### Weiteres Vorgehen

Da nach der Analyse klar ist, dass es keine Cross Site Scripting Probleme vorhanden sind, müssen wir diesbezüglich nicht viel unternehmen. Ein Teil von Cross Site Scripting wird auch durch die File Manipulation abgedeckt, bei welcher Script Code in Files abgespeichert werden könnte. Da das Vorgehen bei File Manipulation aber schon definiert ist, können wir diese Schritte hier weglassen.

## Encoding

Je nach dem wie ein File codiert ist, können einige Zeichen nicht richtig interpretiert werden. Ein grosses Problem sind immer wieder die Umlaute "ä", "ö" und "ü". Deshalb ist es wichtig gerade bei deutschen und internationalen Applikationen Encoding-Standards zu benutzen, welche die verschiedensten Zeichen anzeigen kann.

### Vorhandene Probleme

Die Applikation kann momentan mit den Umlauten nicht umgehen und zeigt dort deshalb falsche Zeichen an in den Files aufgaben.inc.php und eigenschaften.in.php.

### Weiteres Vorgehen

Die Files, welche falsch encoded (UTF-8) sind, müssen in konvertiert werden in ein anderes Code-Format. In dieser Applikation sind folgende Files betroffen:

* aufgaben.inc.php
* eigenschaften.in.php

## Penetrationtests

OWASP Zap durchführen

## Verschlüsselung

Die Verschlüsselung von sensitiven Daten in eine Applikation ist ein sehr wichtiger Punkt der Sicherheit. Ist diese Verschlüsselung z.B. bei Passwörtern nicht vorhanden, schwach oder veraltet, ist es sehr einfach ein Konto zu knacken. Deshalb ist es wichtig, dass man die sensitiven Daten verschlüsselt und eine passende Verschlüsselungstechnik wählt. Dabei ist nicht nur die Verschlüsselungstechnik wichtig, sondern auch der Gebrauch von Salt und Pepper. So werden die sensitiven Daten optimal geschützt.

### Vorhandene Probleme

Die Passwörter in der Datenbank von Schlosslauf werden verschlüsselt abgespeichert. Jedoch haben wir das Problem, dass falls ein Angreifer Zugriff auf die Daten erhält, er die Passwörter sehr schnell geknackt hat, da die md5-Verschlüsselung sehr bekannt ist.   
Beim Eingeben des md5-Wertes im Browser wird einem direkt der entschlüsselte Wert geliefert. Dafür braucht man also nicht mal mehr ein Entschlüsselungsprogramm.

### Weiteres Vorgehen

Damit die Passwörter der Applikation etwas sicherer sind, muss deshalb eine andere Verschlüsselungstechnik verwendet werden. Am besten eine eher neue, noch nicht so bekannte. So ist die Wahrscheinlichkeit, dass bereits viele Rainbow-Tables existieren, etwas kleiner. Ausserdem sollte zum Passwort ein Salt-Wert hinzugefügt werden und ein Pepper definiert sein.

Deshalb der Vorschlag anstelle der md5-Funktion in PHP die password\_hash-Funktion zu verwenden. Sie ist sicherer als md5, jünger und bietet gleich noch die Möglichkeit einen Salt hinzuzufügen.

# Sicherheitsrisiken

# Sicherheitsverbesserungen

# Resultate

# Quellen

|  |  |
| --- | --- |
| Name | Link |
| Rips | https://sourceforge.net/projects/rips-scanner/?source=typ\_redirect |
|  |  |
|  |  |
|  |  |
|  |  |