Университет ИТМО, факультет программной инженерии и компьютерной техники
Двухнедельная отчётная работа по «Информатике»: аннотация к статье
Дата прошедшей лекции: 11.10.2022 Номер прошедшей лекции: 3 Дата сдачи: 25.10.2022

Выполнил(а) Карташев В. С., № группы *P3131*, оценка _____
Фамилия И.О. студента — не заполнять

Название статьи/главы книги/видеолекции
Как регулярное выражение может стать причиной ReDoS-уязвимости?

ФИО автора статьи (или e-mail)	**Дата публикации (не старше 2019 года)**	**Размер статьи (от 400 слов)**
@AndreyMoskalew	"03" ноября 2022 г.	1414

Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)
shorturl.at/kJKO4

Теги, ключевые слова или словосочетания
redos, regex, уязвимость, регулярные выражения, катастрофический возврат, dos…

Перечень фактов, упомянутых в статье
1. Цель ReDoS-атаки – помешать работе приложения с помощью неэффективного регулярного выражения.
2. 1 тип. В приложение передается строка, содержащая опасный паттерн. Далее эта строка используется в качестве регулярного выражения, что приводит к ReDoS.
3. 2 тип. В приложение передается строка определённого формата. Далее эта строка оценивается уязвимым регулярным выражением, что приводит к ReDoS.
4. Существует строка, которую можно было бы сопоставить с обоими подвыражениями (строку *1111* можно сопоставить как с шаблоном *\d+*, так и с *(-?\d+)**).

Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)
Способы защиты:
1. Добавить ограничение на время обработки строки регулярным выражением.
2. Использовать атомарные группы *(?>...)*.
3. Переписать регулярное выражение, заменив небезопасное подвыражение на безопасный аналог.

Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)
1. Регулярное выражение уязвимо к катастрофическому возврату, если оно содержит хотя бы одно подвыражение, из-за которого может возникнуть большое количество вариантов сопоставлений.
2. Регулярные выражения могут стать причиной уязвимости к ReDoS-атаке, цель которой - остановить или сильно замедлить приложение.
3. Регулярное выражение уязвимо к катастрофическому возврату, если содержит хотя бы одно уязвимое подвыражение, из-за которого может возникнуть большое количество вариантов сопоставлений.

Ваши замечания, пожелания преподавателю *или* анекдот о программистах[1]
В поезде едут 3 юзера и 3 программиста.
У юзеров 3 билета, у программистов 1. Заходит контроллер.
Юзеры показывают билеты, программисты прячутся в туалет. Контроллер стучится в туалет, оттуда высовывается рука с билетом. Программисты едут дальше.
На обратном пути. У юзеров 1 билет, у программистов ни одного. Заходит контроллер.
Юзеры прячутся в туалет. Один из программистов стучит, из туалета высовывается рука с билетом.
Программисты забирают билет и прячутся в соседний туалет. Юзеров ссаживают с поезда.
Вывод — не всякий алгоритм, доступный программисту, доступен юзеру.

[1] Наличие этой графы не влияет на оценку