Практическая работа 6

DEFER\_AFTER\_NULL.pdf

Задание



Анализ кода

Этот Warning анализатора относится к **CWE-476: NULL PointerDereference**.

Рассмотри данные фрагмент кода. Указатель *inj\_cond* на 1246 строчке сравнивается с NULL. Если он не равен NULL, то у *inj\_cond* вызывается метод *fix\_fields(thd, 0).*

Далее в следующем условии (не вложенном) на 1249 строчке вызывается функция *inject\_cond\_into\_where* и в нее передается результат вызова метода *copy\_andor\_structure(thd)* у *inj\_cond*. В этот момент происходит разыменование указателя.

Скорее всего, анализатор подумал, что это разыменование происходит в else блоке предыдущего условия. В таком случае *inj\_cond* был бы равен NULL.

Но на самом деле это не так. Однако ошибка все равно присутствует. Потенциально, *inj\_cond* может быть равен NULL. В момент разыменования на 1249 строчке.

Null Pointer Dereference может привести к следующем уязвимостям:

1. Неопределенное поведение: Разыменование NULL-указателя может привести к неопределенному поведению программы, что делает её уязвимой для атак. В большинстве случаев это вызывает исключение, приводящее к сбою приложения или системы.

2. Увеличение привилегий: Исследования показывают, что разыменование NULL-указателей в ядре Linux может быть использовано для повышения привилегий. Например, команда Google Project Zero разработала метод, позволяющий эксплуатировать такие уязвимости для выполнения произвольного кода на уровне ядра, что потенциально может привести к полному контролю над системой.

3. Отказ в обслуживании (DoS): В некоторых случаях разыменование пустого указателя может вызвать сбой приложения или системы, что приводит к отказу в обслуживании. Это может быть использовано злоумышленниками для временной недоступности сервиса или приложения.

Предложение по исправлению

На 1249 строчке нужно добавить проверку на NULL перед разыменованием. То есть:



